NAVAL POSTGRADUATE SCHOOL

MONTEREY, CALIFORNIA

THESIS

WIND ASSESSMENT FOR AERIAL PAYLOAD DELIVERY SYSTEMS USING GPS AND IMU SENSORS

by

Matthew D. O'Brian

September 2016

Thesis Advisor:	Oleg A. Yakimenko
Second Reader:	Fotis A. Papoulias

THIS PAGE INTENTIONALLY LEFT BLANK

REPORT DOCUMENTATION PAGE			Form Approved OMB No. 0704–0188
Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instruction, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188) Washington, DC 20503.			
1. AGENCY USE ONLY ***(Leave blank)***	**2. REPORT DATE** September 2016	**3. REPORT TYPE AND DATES COVERED** Master's thesis	
4. TITLE AND SUBTITLE WIND ASSESSMENT FOR AERIAL PAYLOAD DELIVERY SYSTEMS USING GPS AND IMU SENSORS			**5. FUNDING NUMBERS**
6. AUTHOR(S) Matthew D. O'Brian			
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Naval Postgraduate School Monterey, CA 93943-5000			**8. PERFORMING ORGANIZATION REPORT NUMBER**
9. SPONSORING /MONITORING AGENCY NAME(S) AND ADDRESS(ES) N/A			**10. SPONSORING / MONITORING AGENCY REPORT NUMBER**
11. SUPPLEMENTARY NOTES The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government. IRB number ____N/A____.			
12a. DISTRIBUTION / AVAILABILITY STATEMENT Approved for public release. Distribution is unlimited.			**12b. DISTRIBUTION CODE**

13. ABSTRACT (maximum 200 words)

Numerous organizations within the Department of Defense have requested research and development efforts to create a lightweight Joint Precision Airdrop System (JPADS) capable of covertly distributing items to austere or contested locations. This mission has many critical challenges, with meteorological estimation near the top of the list due to a ram-air parachute's high susceptibility to environmental forces. Computer-based modeling of environmental conditions is extremely difficult due to the chaotic and often unpredictable interactions of environmental factors and the surrounding topography, so bench tests, flight tests, and the post-processing of the resultant test data were the research methods used in development of this thesis. Ultimately, this thesis presents two models for winds aloft prediction capable of presenting an increased fidelity solution. Both methods were field tested and could be used in JPADS guidance, navigation, and control algorithms.

14. SUBJECT TERMS JPADS, aerial delivery, commercial off-the-shelf, parachute control, wind estimation, quaternions, dynamic equations, kinematic equations, MATLAB			**15. NUMBER OF PAGES** 89
			16. PRICE CODE
17. SECURITY CLASSIFICATION OF REPORT Unclassified	**18. SECURITY CLASSIFICATION OF THIS PAGE** Unclassified	**19. SECURITY CLASSIFICATION OF ABSTRACT** Unclassified	**20. LIMITATION OF ABSTRACT** UU

NSN 7540–01-280-5500 Standard Form 298 (Rev. 2–89)
Prescribed by ANSI Std. 239–18

THIS PAGE INTENTIONALLY LEFT BLANK

Approved for public release. Distribution is unlimited.

WIND ASSESSMENT FOR AERIAL PAYLOAD DELIVERY SYSTEMS USING GPS AND IMU SENSORS

Matthew D. O'Brian
Lieutenant Commander, United States Navy
B.S., United States Naval Academy, 2000
M.S., Boston University, 2008

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN MECHANICAL ENGINEERING

from the

NAVAL POSTGRADUATE SCHOOL
September 2016

Approved by: Oleg A. Yakimenko
Thesis Advisor

Fotis A. Papoulias
Second Reader

Garth Hobson
Chair, Department of Mechanical and Aerospace Engineering

THIS PAGE INTENTIONALLY LEFT BLANK

ABSTRACT

Numerous organizations within the Department of Defense have requested research and development efforts to create a lightweight Joint Precision Airdrop System (JPADS) capable of covertly distributing items to austere or contested locations. This mission has many critical challenges, with meteorological estimation near the top of the list due to a ram-air parachute's high susceptibility to environmental forces. Computer-based modeling of environmental conditions is extremely difficult due to the chaotic and often unpredictable interactions of environmental factors and the surrounding topography, so bench tests, flight tests, and the post-processing of the resultant test data were the research methods used in development of this thesis. Ultimately, this thesis presents two models for winds aloft prediction capable of presenting an increased fidelity solution. Both methods were field tested and could be used in JPADS guidance, navigation, and control algorithms.

THIS PAGE INTENTIONALLY LEFT BLANK

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF ACRONYMS AND ABBREVIATIONS

3D	Three Dimensional
ADS	Aerial Delivery System
AGL	Above Ground Level
AGU	Airborne Guidance Unit
AIAA	American Institute of Aeronautics and Astronautics
ATC	Air Traffic Control
CEP	Circular Error Probable
CG	Center of Gravity
CIRPAS	Center for Interdisciplinary Remotely-Piloted Aircraft Studies
DOD	Department of Defense
DPI	Desired Point of Impact
FAA	Federal Aviation Administration
GNC	Guidance, Navigation, and Control
GPS	Global Positioning System
GUI	Graphic User Interface
IED	Improvised Explosive Device
JPADS	Joint Precision Airdrop System
L/D	Lift over Drag Ratio
MANPADS	MAN-Portable Air-Defense System
MOA	Military Operating Area
PADS	Precision Aerial Delivery System
R/C	Radio-Controlled
UAV	Unmanned Aerial Vehicle
VTOL	Vertical Take-Off and Landing

THIS PAGE INTENTIONALLY LEFT BLANK

ACKNOWLEDGMENTS

I would like to thank Professor Oleg Yakimenko for his untiring enthusiasm and willingness to give advice and guide me in the right direction. Also, I would be remiss without extending a word of thanks to Professor Fotis Papoulias for accepting the responsibility of a second reader for a student outside your department. To the professionals at ARCTURUS UAV, thank you for providing the means to get Snowflake airborne and enduring some hot times in the Paso Robles sun. I would also like so like to say thank you to Drew Hall. Drew and I charted parallel courses and endured many long hours in the sweatshop, so thank you for your sacrifices and friendship over the past few quarters. Finally, to my wife, Amy, and the little ones, thank you for your steadfast belief in me and your unrelenting support through the many hours I spent away from home. My family is my rock and none of this would have been possible without you!

THIS PAGE INTENTIONALLY LEFT BLANK

I. INTRODUCTION AND PROBLEM FORMULATION

Parachutes come in all different shapes, sizes, and colors, and for every stylistic variation there is an equally diverse set of employment methods and objectives. For instance, parachutes are used as safety mechanisms in ejection seats for high-performance aircraft and as deceleration devices for spacecraft re-entering Earth's atmosphere. Parachutes are by no means a new product, but the way that parachutes are used has changed drastically over the last few centuries. To that end, in the last 30 years there has been an interest in creating autonomous, precision guided parachute delivery systems capable of accurately delivering payloads for both military and civilian applications. However, there are many challenges associated with the proposal of autonomous control for parachutes; and perhaps the most influential challenge is the impact of environmental factors upon the parachute system. Thus, it is vital to know the prevailing atmospheric conditions, especially the winds, in order to accurately deliver a self-guided parachute to a specific location. However, before delving into the realm of wind estimation, a foundation of historical parachute applications and current employment methods must be understood.

A. HISTORICAL DESIGN AND APPLICATIONS

The word *parachute* is derived from the French and Greek meaning to "shield against falling" [1]. Although little documentation exists, it is believed that Chinese acrobats in the early 14^{th} century were among the first to employ parachutes to slow their descent to the ground. However, the first documented depiction of a parachute is credited to Leonardo da Vinci [1]. In Figure 1, da Vinci illustrates a human suspended from a pyramid-shaped parachute design.

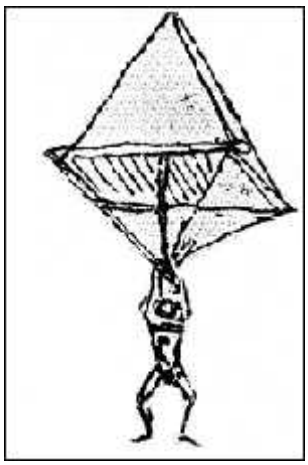

Figure 1. Early Parachute Sketch by Leonardo da Vinci. Source: [1].

It is unclear whether da Vinci ever attempted to fabricate his design or if he simply sketched his idea as a proof of concept. Note the rigid frame that da Vinci depicted at the bottom of the pyramid. A few similar inventions were built and tested in the decades to come, but a non-rigid canopy more closely related to present-day parachutes was not utilized until October 22, 1797. On that day, an inventor and adventurer named André-Jacques Garnerin jumped from a balloon poised 975 meters above the city of Paris, France. As a result, he became known as the "first to design and test parachutes capable of slowing a man's fall from a high altitude" [2].

For a little over a century, inventors and adventure enthusiasts continued to toy with the concept of parachuting, but the U.S. military paid little to no attention to these daredevils. In fact, the first U.S. military airdrop of supplies did not occur until the end of World War I, and since this event occurred at the end of the war, there was no further development of military airborne delivery systems. However, other nations like Russia developed in-depth contingency operations for parachutes and as a result, "the Russian Red Army was the first to airdrop soldiers and weapons with their crew" [3]. The U.S. Army took note of the Russian successes, resulting in the creation of the 82nd and 101st Airborne Divisions, both of which saw extensive operations on the European front during World War II. During the infamous D-day operations, U.S. paratroopers made

history when over 13,000 American paratroopers jumped from hundreds of airplanes under the cover of darkness to kick off the Normandy invasion. From that day forward, nearly all battle plans have been created with some degree of paratrooper involvement. It is noteworthy to mention that the multitude of parachute canopies depicted in Figure 2 represents only a small percentage of the total paratroopers that jumped into the French countryside on D-day.

Figure 2. Paratroopers Jump into Conflict on D-day during the Normandy Invasion on June 6, 1944. Source: [4].

B. EXTENSIONS OF OPERATIONS

Parachute delivery of soldiers yields an unmatched degree of flexibility to combatant commanders. Similarly, the same benefits of airborne troop delivery can be gained by airborne weapon system delivery and re-supply. In fact, the concept of parachute delivery systems can be attached to a plethora of diverse applications. For example, parachutes can be used to deliver humanitarian relief in the wake of natural disasters when other delivery methods are too risky, slow, or difficult. Additionally, parachutes can be used to deliver water, food, and ammunition to troops operating covertly in austere locations. It is important to note that parachute delivery systems are not solely utilized in military

applications. Major companies, like Amazon, could benefit from a precision aerial delivery vehicle to help meet delivery timeline goals when customers live in remote locations. However, in order to function in all these capacities, the parachute delivery system must have a higher degree of precision than simply pushing the package out the back of an airplane and hoping for the best. As a result, over the past 25 years there has been significant research and development in a parachute modification and enhancement called the Precision Aerial Delivery Systems (PADS).

C. FIELDED PRECISION AERIAL DELIVERY SYSTEMS

The U.S. Army and numerous civilian organizations have fielded dozens of PADS during the past couple decades. The “majority of PADS use ram-air parachutes” instead of round parachutes “because of their relatively high glide capability and controllability” [5]. Figure 3 depicts a typical PADS layout complete with ram-air parachute, control unit, and payload.

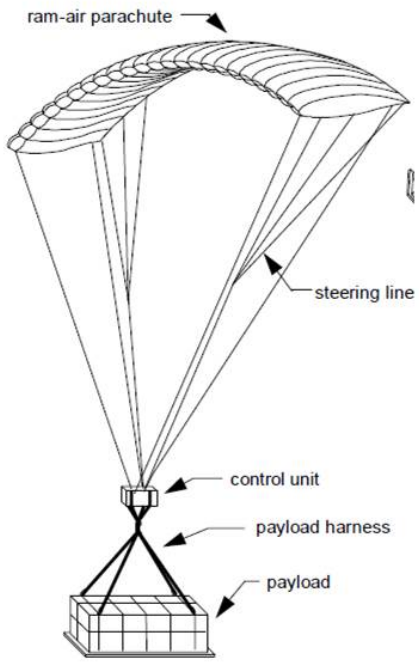

Figure 3. Inflated Ram-Air Parachute with Payload. Source: [6].

In Lindgard's paper, he argues that "because of its high glide capability and its controllability the ram-air parachute offers considerable scope for the delivery or recovery of payloads to a point by automatic control linked to a guidance system" [6]. In the same paper, Lindgard stresses to the reader that the use of PADS is a cost-effective and potentially life-saving tactic because PADS can be released from aircraft at high altitude. In contrast, previous "precision" delivery methods required the delivery aircraft to fly low and slow making it susceptible to enemy small arms fire and Man-Portable Air Defense Systems (MANPADS). The low altitude drop was required in order to increase delivery accuracy by reducing the payload and parachute's exposure to winds aloft. However, by using the PADS, "a parachute with glide and a control system can compensate for inaccuracies in drop point and wind" [6].

As a direct result of the reduced risk benefits, the U.S. Army and U.S. Air Force teamed up to create the Joint Precision Airdrop Systems (JPADS). According to the U.S. Army Natick Soldier Research and Development Center, JPADS is an "integration of Army Guided, High Altitude Parachute System and the Air Force Precision Airdrop Mission Planning System" [7]. JPADS provides the following list, bringing together:

- Gliding parachute decelerators
- GPS-based guidance, navigation, and control
- Weather data assimilation and airdrop mission planning tool
- Wireless gate release system and complimentary programs [7]

One unique characteristic of JPADS is the ability to use different parachutes depending on the payload size. Initially the program began with six broad payload categories, but this has since been reduced to five categories. Table 1 depicts the current JPADS category titles along with their approximate payload delivery capabilities.

Table 1. JPADS Categories and Approximate Payloads. Source: [5].

JPADS CATEGORY NAME	APPROXIMATE PAYLOAD
Micro-lightweight (ML)	~5-70 kg (10-150 lb)
Ultra-lightweight (UL)	~100-300 kg (250-700 lb)
Extra-lightweight (XL)	~300 kg- 1.1 tons (700-2,400 lb)
Lightweight (L)	~2.3-4.5 tons (5,000-10,000 lb)
Medium weight (M)	~4.5-19 tons (10,000-42,000 lb)

In field testing, the JPADS has performed adequately. “As of 2012, JPADS 2K [the XL category system] featured circular error probable (CEP) touchdown accuracy of 150 m (490 ft) and JPADS 10 k [L category system] exhibited 250 m (820 ft) CEP” [5]. To the untrained eye, the delivery of a multi-ton payload within about 200 meters seems pretty good. However, the data is somewhat tainted when CEP is used as an accuracy metric. In other words, CEP looks good on paper, because many readers do not know exactly what it means. Driels defines CEP as “the radius of a circle, centered on the Desired Point of Impact (DPI) such that 50% of the impact points lie within it” [8]. Thus, in the JPADS 2K and JPADS 10K data mentioned earlier, only 50% of the payloads were delivered within 490 ft and 820 ft respectively. There is no statistical description of the other 50% of trials, which is very significant because they might have been one foot outside the CEP or 1 mile, yet no mention of this is given to the reader.

Another potential problem with the current JPADS system is the cost. Currently fielded systems’ costs vary based on their size category, but they typically range between $68,000 and $100,000. Consequently, the troops on the ground must put themselves at risk in order to recover these non-expendable units so the JPAD unit can be brought back to a base, re-purposed, and re-used. Therefore, there exists a need for an expendable, commercial off-the-shelf unit, which can deliver superior results without the need for recovery.

D. FLEET APPLICATIONS

At first glance, it seems as though a Snowflake is designed with the intention of resupplying forward operating ground forces as its primary mission. However, there are many nautical applications where Snowflakes can be used to ensure battlespace dominance. Take for instance, a scenario where the Navy wants to deploy acoustic sensors in a contested region. In this case, multiple Snowflakes could be equipped with the acoustic payload and each Snowflake could be programmed to glide to a different location. The result is a covertly inserted network of acoustic sensors, which were deployed with minimal risk. Piggy-backing on the multiple sensor network theory, the use of multiple Snowflakes could also be used to covertly install communication nodes on building rooftops. Consequently, when ground forces move in to the urban environment, they already have a secure communications network established. A final example of Snowflake's versatility is a weaponized version. When a 500-lb bomb is too large, the Snowflake could deliver a small warhead to a precise location. In this capacity, the weaponized warhead is well suited for neutralizing improvised explosive devices (IEDs) or mines, both at sea and on the land. Therefore, the host of Snowflake applications begins and ends with the imagination of the operational planners.

E. THESIS OBJECTIVE AND ORGANIZATION

The main objective of this thesis is to explore multiple methods for wind estimation based on field experimentation with a miniature aerial payload delivery system called Snowflake. Prior to addressing the wind estimation techniques, a foundation of general parachute principles must be discussed. The brief historical and current operations introduction is followed by a review of the Snowflake components in Chapter II and equations of motion for a parachute and payload in Chapter III. Following Chapter III, the ensuing chapter discusses the experimentation procedures used for the wind estimation methods and most importantly a description of the prevailing topography and atmospheric conditions

of the test site at Camp Roberts, CA. A basic understanding of all this information is necessary to lay the foundation for the two wind estimation methods developed and presented in Chapter V. Next, Chapter VI is devoted to an in-depth analysis of experimental test data. A comparative analysis of the two wind estimation methods is also provided in Chapter VI. Finally, this thesis finishes with some concluding remarks provided to capture the contributions of the wind estimation methods for an aerial payload deliver system.

II. BLIZZARD SYSTEM COMPONENTS

The exorbitant cost per unit of JPADS coupled with the undue stresses and endangerment of U.S. military members is uncalled-for when commercial off-the-shelf components are available that poses less risk while ultimately improving cost effectiveness. The cost of individual system components was very high 20 years ago, when government-funded research and development of unmanned systems began to flourish. However, today, hundreds of small businesses and large corporations alike are producing highly sophisticated system components for fractions of the cost. The Snowflake design scheme seeks to capitalize on these affordable, readily available, and technologically advanced apparatuses. This chapter highlights some of the individual components that were experimented with and tested in order to find the best combination of technology and cost effectiveness. First, the chapter begins with a description of the Snowflake system, followed by an overview of two different autopilots used in the Snowflake's airborne guidance unit (AGU). The chapter concludes with the presentation of the deployment platform, the Arcturus UAV. When combined, the combination of the Snowflake ADS and the Arcturus UAV make up the Blizzard system.

A. SNOWFLAKE AERIAL DELIVERY SYSTEM

The ram-air parachute is the preferred flight system for Snowflake because of its versatility and maneuverability. Ram-air parachutes are used around the world by skydiving enthusiasts and professional parachute demonstration teams. The alternative choice for a parachute is the circular canopy. However, circular canopies lack the controllability required to be considered as a suitable choice for a Snowflake aerial delivery system (ADS). Thus, the decision was made to use ram-air parachutes similar to the one pictured in Figure 4 as the aerodynamic deceleration device. Two different sized ram-air parachutes were used in conjunction with Snowflake experimentation.

The first parachute measured 1.0 m^2 and the second was 1.5 m^2. Ultimately, when dealing with ram-air parachutes, it is crucial to have an understanding of the aerodynamic characteristics and the importance of rigging angles.

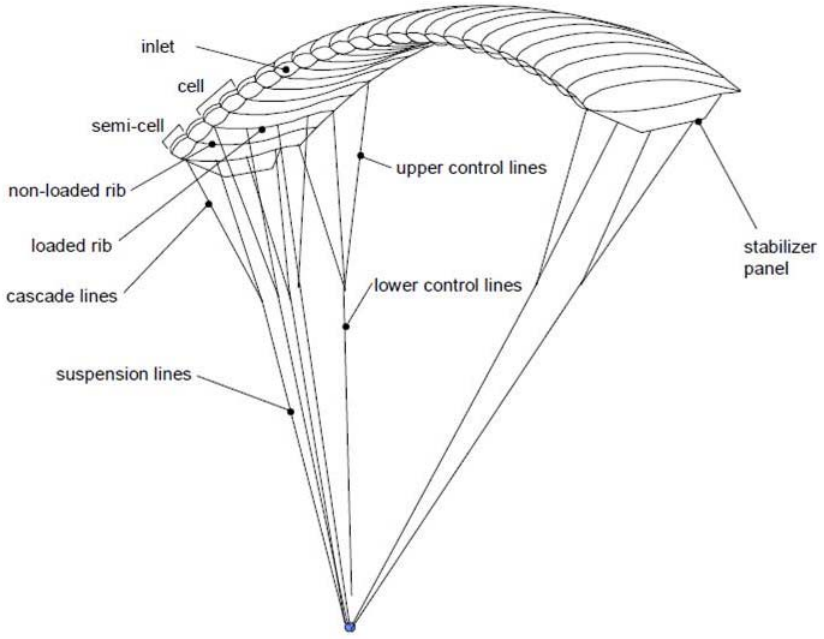

Figure 4. Ram-air Parachute Components. Source: [6].

1. Aerodynamics Considerations

"Ram-air gliding parafoils caused a real revolution in terms of accuracy when they were introduced in skydiving in the early 1970s ending the era of round canopies" [5]. Unlike a round canopy, the ram-air parachute has multiple cells, which channelize the air and allow it to flow through the canopy. Consequently, "once inflated, a ram-air parachute is essentially a low-aspect-ratio wing, and thus conventional wing theory is applicable" [5].

In general, wings are classified into two broad categories based on the way they bend relative to an imaginary line perpendicular to the direction of airflow. Therefore, when looking at the front view of a wing,

> the angle between the chord-line plane of [the] wing with the “xy” plane is referred to as the wing dihedral. If the wing tip is higher than the xy plane, the angle is called positive dihedral or simply dihedral, but when the wing tip is lower than the xy plane, the angle is called negative dihedral or anhedral. [9]

The difference between the dihedral and anhedral wing is best summarized with a drawing like the one in Figure 5.

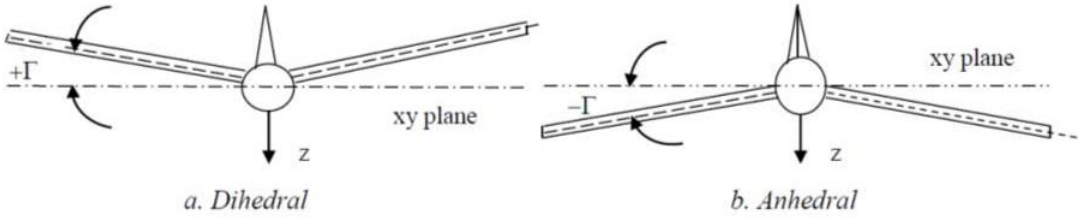

Figure 5. Dihedral vs. Anhedral Wing Comparison. Source: [9].

The wing style employed on ram-air parachutes, and Snowflake, most closely resembles the anhedral type as indicated by the downward bend of the parachute when properly inflated. Figure 6 is a photo taken during field experiments in Camp Roberts, CA, that clearly illustrates the downward curvature of the ram-air parachute when it is properly inflated and the support lines are under tension from a payload.

Figure 6. Properly Inflated Anhedral Design Snowflake Parachute.

An anhedral arc is “slightly detrimental to the wing’s lifting properties while the drag coefficient for a given angle of attack remains as that for a flat wing” [5]. Furthermore, the anhedral characteristic of a ram-air parachute is a direct function of the suspension lines pulling down on the canopy. In Snowflake’s case, the suspension lines connect the payload to the parachute, so the slight decrease in lift created by the anhedral wing is unavoidable. Additionally, the suspension lines are of critical importance for the anhedral shape of the wing, the overall stability of the system, and production of the desired Lift over Drag ratio (L/D).

2. Rigging Angles

Proper rigging angle is one of the most critical design specifications required in order for a ram-air parachute to achieve coordinated flight. The term rigging angle in general means “positioning the payload, and hence the center of gravity (CG) of the system, such that the equilibrium attitude is at the required angle of attack” [5]. Ultimately, the length of the suspension lines is the critical metric that determines whether the rigging angle is correct. It is imperative to ensure acute attention to detail when rigging the parachute because even the smallest deviation from the required measurement can induce instabilities and

prohibit the system from returning to its equilibrium point after a disruptive force is encountered. Figure 7 shows a properly rigged ram-air parachute along with the associated angles. In Figure 7, the angle μ is the rigging angle for the parachute.

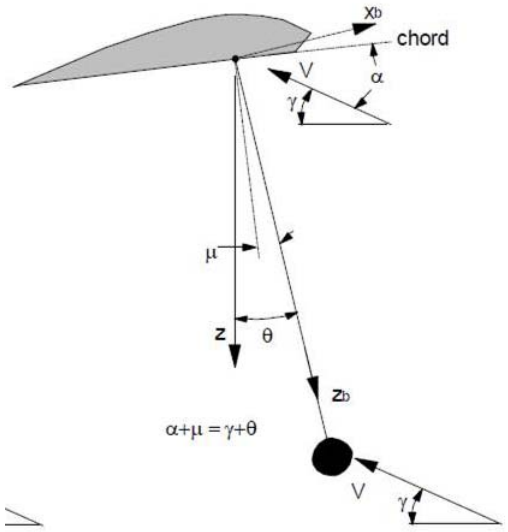

Figure 7. Important Angles Required Ensuring Parachute Stability.
Source: [6].

Ensuring the proper rigging angle for the Snowflake system proved to be a daunting and frustrating task during the construction and field testing phases. In order to properly rig the parachute, lines had to be trimmed and very small knots had to be tied in precise locations in order to achieve a stable platform. It was noted during experimentation that a small discrepancy in rigging angle often led to improper canopy inflation and the degradation of the Snowflake's desired L/D ratio.

3. Control Lines

Proper rigging of the control lines is another task that requires steadfast attention to detail during the construction phase in order to produce predictable and reliable ram-air parachute flight control. The control lines attach to the

training edge of the ram-air parachute and are actuated by extension or retraction of the line. For instance, a retraction on the left or right control line will induce a left turn or right turn respectively. Also, retracting both control lines simultaneously will result in a flare maneuver used to slow the rate of descent. Through experimentation with Snowflake, it was determined that a difference in control line length between left and right sides greater than two inches resulted in a tight spiral and a significant loss of lift. This experiment clearly demonstrated that even a small difference in line lengths could result in significant changes to the flight characteristics of the Snowflake system.

B. AIRBORNE GUIDANCE UNIT

The autopilot is analogous to the central nervous system of the Snowflake. In order for an autopilot to be effective, it must be capable of collecting sensor data, determine the pose of the vehicle, implement a guidance and control algorithm, and record everything it "sees" in order to recreate the flight during post-mission analysis. A description of the two autopilots utilized during field experimentation is presented in this section.

a. Pixhawk

"Pixhawk is an advanced autopilot system designed by the PX4 open-hardware project and manufactured by 3D Robotics" [10]. Internally, it employs two redundant accelerometers and gyroscopes along with a barometer. There is also a multitude of external sensors that may be connected such as a Global Positioning System (GPS) antenna, telemetry transmitters, airspeed sensor, and radar altimeter. [10] As Seen in Figure 8, all of the external sensors and equipment are connected to the Pixhawk via numerous plug and play serial interface ports on the top of the device.

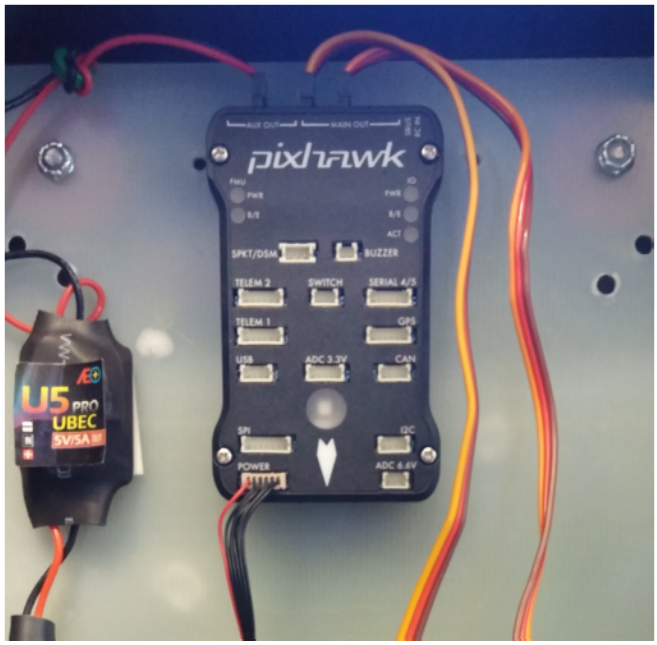

Figure 8. 3DR Pixhawk Autopilot Mounted in a Snowflake.

Another extremely useful benefit of the Pixhawk autopilot is its intuitive Mission Planner software. This free software package includes an intuitive Graphic User Interface (GUI) for on-the-fly control adjustments along with a high fidelity diagnostic layout. The GUI, seen in Figure 9, strongly resembles most computer-based flight simulator programs. The cost of the Pixhawk system employed on Snowflake was approximately $250.

Figure 9. Mission Planner GUI Screen Shot (Not a Snowflake Flight). Source: [11].

At first glance, and through the first six months of experimentation, the Pixhawk seems like the perfect autopilot for Snowflake. However, after a few experimental drops, numerous faults began to arise in the GPS, accelerometers, and gyroscopes. It was determined that the Pixhawk and its sensors were not robust enough to absorb the shock forces of the canopy opening and the eventual impacts with the ground. As a result, the Pixhawk autopilot and its sensor suite were removed from the Snowflake AGU.

b. X-Monkey

The autopilot chosen to replace the Pixhawk in Snowflake was the X-Monkey. In contrast to Pixhawk, the X-Monkey is far less user friendly and does not have nearly the same amount of online, open-source software programmers. Instead, the X-Monkey system is created and produced by Matt Ryan, the founder of Ryan Mechatronics. Thus, what X-Monkey lacks in online, open-source software programming, it quickly makes up for with direct contact with the

creator. Additionally, “the X-Monkey navigation platform integrates the latest ARM Cortex processor with a high performance GPS receiver, barometric pressure sensor, rate, acceleration and magnetic sensors, and micro SD card logging” [12]. The X-Monkey’s sensor suite interface, pictured in Figure 10, is far less user-friendly and lacks the plug-and-play versatility offered by the Pixhawk autopilot. The cost of the X-Monkey system employed on Snowflake was $329.

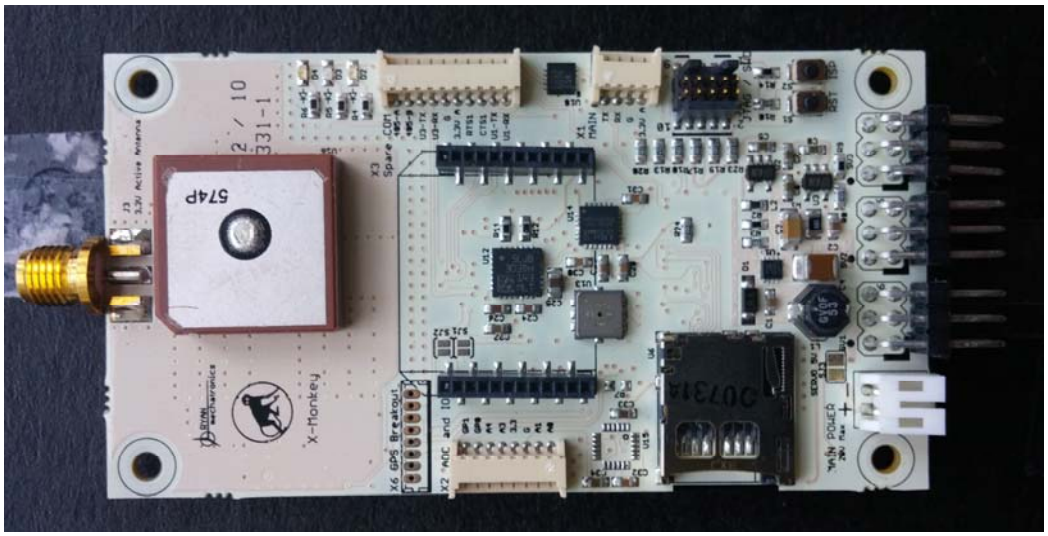

Figure 10. Ryan Mechatronics’ X-Monkey Autopilot.

Unfortunately, there are a few major drawbacks to using the X-Monkey autopilot system. For instance, it does not have an elaborate mission planner software application and the autopilot requires an in-depth knowledge of coding in the C++ language in order to configure any of the ports or incorporate any guidance and navigation algorithms. However, the X-Monkey is physically more robust than the Pixhawk and has not experienced any failures due to the opening canopy shock or ground impact.

C. ARCTURUS UAV

ARCTURUS UAV (unmanned aerial vehicle) is a small unmanned aerial vehicle manufacturing company based in Central California offering multiple UAV configurations that can be tailored to mission specific requirements. As seen in

Figures 11 and 12, the company's flagship is a catapult launched, medium range aircraft constructed of lightweight composite materials. The T-20 has a 5.3 meter wingspan and can remain airborne between 10 and 20 hours based on payload weight. It has the ability to carry payloads internally in its fuselage or externally the wings using hard points.

Figure 11. ARCTURUS T-20 in Flight. Source: [13].

During Snowflake testing, the T-20 used hard points located beneath the wings to carry the Snowflake to altitude, release the device, and deploy the parachute via static line release.

Figure 12. ARCTURUS T-20 on the Catapult with Two Snowflakes Mounted under the Wings.

ARCTURUS UAV also has a second model in production, which is called the JUMP20. As pictured in Figure 13, this aircraft is based on the T-20 frame but employs a vertical takeoff and landing (VTOL) capability for operations in austere locations where a catapult and belly landing are not capable or noise abatement is required for to maintain covert operations security. The JUMP 20 employs four electric motors, which drive fixed pitch propellers enabling the aircraft to take off like a quadcopter, transition to forward flight like an airplane, and recover using the vertical quadcopter configuration.

Figure 13. ARCTURUS JUMP20 with Quadrotors Engaged for Vertical Takeoff. Source: [13].

Snowflake was employed successfully on the JUMP20 variant after some modifications were made to the Snowflake parachute bag ensuring the downwash of the quadrotors would not lead to pre-deployment of the parachute during critical phases of flight (takeoff and landing).

III. KEY CONCEPTS OF MOTION MODELING

As described in [5], a ram-air parachute can be modeled as a low-aspect-ratio wing. This concept helps to simplify the equations of motion for the parachute, but there is a lot more information that must be captured in order to fully understand and accurately model the flight of a Snowflake. Thus, it is imperative to create a working knowledge of the dynamic and kinematic equations of motion for a Snowflake. Additionally, this chapter will briefly touch on the important topic of quaternions, which is how the X-Monkey autopilot records the orientation of the Snowflake.

A. ADS DYNAMICS

In its most basic form, dynamics is simply the study of the effects on a body in motion when acted on by a force or torque. For parachute delivery systems, “dynamical analysis is important in that it can indicate conditions under which steady glide is rapidly achieved, and conditions where dynamic instability may occur or transient motions are only lightly damped” [6]. In order to reduce some of the complexity of a parachute delivery system, the parachute, all lines, and the payload will all be treated as one rigid body. This assumption is valid as long as the suspension lines are the proper length and the payload flies in equilibrium below the parachute without too many egregious oscillations. This basic assumption is depicted in Figure 14.

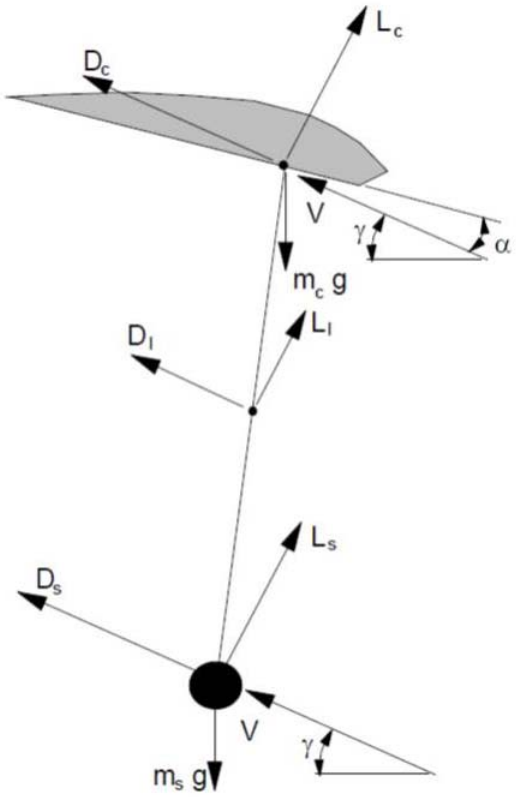

Figure 14. Parachute and Payload Forces and Moments: Source: [6].

Lingard proposed the following definition for the moment *M* about the parachute [6]:

$$M = M_{c/4} + R\left[L_s \sin(\alpha+\mu) - D_s \cos(\alpha+\mu)\right] + \frac{R}{2}\left[L_l \sin(\alpha+\mu) - \mathrm{D}_l \cos(a+\mu)\right] - m_s g R \sin(\theta)$$

In this equation, $M_{c/4}$ is the pitching moment of the canopy about 25% chord point; L_l is the lift force acting on the suspension lines; L_d is the lift force acting on the payload; D_l is the drag force acting on the suspension lines; D_s is the drag of the payload; m_s is the mass of the payload; g is the acceleration due to gravity; R is the distance from the 25% canopy chord point to the payload; and μ is the rigging angle as defined in Figure 7.

From Lingard, one can ascertain the complexity of a rigid body, parachute/ payload system. This one equation does not even begin to graze the surface of

this topic, but is demonstrative of the attention to detail that is required in order to accurately model the dynamical relationships required for simulation and modeling. The purpose of this section is to introduce the topic of dynamics, and a much more in depth review is available in both [6] and [5].

B. ADS KINEMATICS

Kinematics has many similarities to dynamics but in its simplest form, kinematics is the study of position, velocity, and acceleration without any reference to the force or torque that produced the motion. In fact, dynamics and kinematics are so similar that many texts use the two words interchangeably, which is not necessarily accurate. Using the assumption that the parachute, all lines, and the payload form a rigid body, the major contributors to the kinematic equation of motion are initial aircraft velocity (which breaks down into a horizontal a vertical component of initial rigid body velocity), gravity, wind, and drag. In order to produce a relatively accurate model of an unguided parachute, one can use a linear drag, point mass model. In [8], Driels surmises that a linear drag model is a "gross simplification of the effects of air resistance on the projectile, ignoring the variation of air density and temperature as the projectile changes altitude." However, in Snowflake's case, initial evaluation drops were commenced at an altitude of approximately 750 meters above ground level (AGL). Using the standard adiabatic lapse rate of a 2°C decrease in temperature for every 304.8 meters (1,000 feet) of altitude gain, this equates to a change of only 4°C over the entire flight and a minimum change to drag due to atmospheric conditions.

As briefly mentioned in Chapter III, Section A, analysis of key flight characteristics such as vertical velocity, time of flight, and lateral displacement requires the problem to be broken down into vertical and horizontal components.

1. Kinematic Equations in a Vertical Plane

As Driels discusses in [8], the analysis of the vertical components begins with Newton's Second Law

$$F = ma$$

A simple substitution of the vertical components of the characteristics previously discussed results in the following equation,

$$mg - C_d v_v = m\frac{dv_v}{dt}$$

In this form, C_d is the linear drag coefficient and v_v is the vertical velocity less any vertical components of wind

Incidentally, the vertical wind velocity component is discussed but not included in the equations because it is typically assumed to be zero unless there is a clear indication of the presence of thermal updrafts or mechanical winds caused by large buildings or topographical feature changes. Through simple manipulation and the use of Laplace transforms as demonstrated in [8] the vertical velocity at any time (t) can be solved by using the following equation:

$$v_v = \left[v_{0v} - \frac{g}{c_0} \right] \exp\left(-c_0 t\right) + \frac{g}{c_0}$$

In the preceding equation, c_0 is equal to Cd / m and v_{0v} is the initial vertical velocity, assumed to be zero if the aircraft dropping the projectile is in strait and level, coordinated flight. Simply integrating the vertical velocity equation and making some substitutions allows for the vertical position (altitude) at any point to be solved for using the following equation:

$$h = h_0 - \left\{ \frac{1}{c_0} \left[v_{0v} - \frac{g}{c_0} \right] \left[1 - \exp\left(-c_0 t\right) \right] + \frac{gt}{c_0} \right\}$$

where h_0 is the initial aircraft altitude in meters AGL. The linear drag, point mass model is by no means a perfect representation of Snowflake's kinematic characteristics, but it will at least provide reasonably accurate information for the vertical velocity and altitude at any given time. Higher fidelity models are discussed by Driels and Yakimenko in [8] and [5], respectively.

2. Kinematic Equations in a Horizontal Plane

In similar fashion, Driels shows that it is not too difficult to develop the horizontal kinematic equations from Newton's Second Law in order to calculate the horizontal displacement of the projectile, also referred to as the range [8]. Therefore, substituting the horizontal components the point mass' flight parameters yields the following equation:

$$-C_d v_h = m\frac{dv_h}{dt}$$

where v_h is the horizontal component of air velocity. In order to simplify the calculations, the assumption is made that the wind velocity is constant throughout the entire flight. Later chapters will discuss and show that this is a gross approximation, but a necessary one in order to reduce the model's complexity. Once again, through the use of Laplace transforms and some minor equation manipulations, a formula for the horizontal velocity at any given time can be represented as:

$$v_h = v_{0h}\exp(-c_0 t)$$

Additionally, the horizontal distance, or range, travelled by the projectile at any time can be calculated using the following equation

$$x = \frac{v_{oh}}{c_0}\left[1-\exp\left(-c_0 t\right)\right]$$

where v_{0h} is the initial horizontal velocity which is approximately equal to the aircraft's airspeed at the time of release.

C. USING QUATERNIONS

The word quaternion by itself yields a connotation of technologically advanced complexity that many people shy away from and wish to avoid. Contrary to this initial instinct, quaternions are not that difficult to understand and have been around for hundreds of years. "In 1843 [Sir William Rowan] Hamilton

invented the so-called hyper-complex number of rank 4, to which he gave the name *quaternion*. Crucial to this invention was his celebrated rule

$$i^2 = j^2 = k^2 = ijk = -1$$

for dealing with the operations on the vector part of the quaternion" [14]. His revolutionary discovery paved the way for the use of quaternions to describe a specific orientation in three-dimensional space (R^3).

In essence, a quaternion is actually a 4-tuple of real numbers typically depicted as

$$q = (q_0, q_1, q_2, q_3)$$

or

$$q = q_0 + \mathbf{i}q_1 + \mathbf{j}q_2 + \mathbf{k}q_3$$

In the second form of the preceding quaternion representation, "q_0 is the scalar part of the quaternion while **q** is called the vector part of the quaternion" [14]. As a result of the quaternion's unique combination of scalar and vector parts, normal linear algebra principles do not apply, so a completely different set of mathematical rules was created and applied to quaternions. The goal of this section is to introduce the quaternion and its benefits; see [14] for a more detailed explanation and derivation of quaternion algebra.

Before quaternions were used to define coordinate rotations and transformations, the robotic and aeronautical worlds used Euler angles. Unfortunately, as aeronautical products grew in complexity and ability, engineers began to run into problems using Euler angles. "Those that design coordinate transformations know well that inherent in every minimal Euler angle rotation sequence in SO(3)—the group whose elements are the Special Orthogonal matrices in R^3—is at least one singularity" [14] known as gimbal lock. A gimbal is a device commonly attached to fixed-position gyroscopes, which permits rotation through the three axes of motion (roll, pitch, and yaw). Gimbal lock occurs when two of the rotational axis point in the same direction resulting in the loss of 1° of

freedom. In aeronautical applications, this most commonly occurs when the aircraft's pitch approaches +/- 90°. From a mathematical standpoint, this produces a singularity in the Euler angle rotation matrices resulting in erroneous aircraft pose indications in the other two axes, roll and yaw. Quaternions do not exhibit the same gimbal lock singularity making them the preferred choice for applications involving coordinate rotation and transformation.

In addition to gimbal lock avoidance, quaternions take up less memory than Euler angles due to their basic structure. As previously discussed, a quaternion only requires four elements as compared to the nine elements required to represent an Euler angle matrix. On the surface, this may not seem like a major savings because the orientation of the system is sometimes captured and recorded one hundred times per second making the smaller structure size extremely beneficial to a small system with limited memory capability. For size comparison, the following is an example of an Euler rotation matrix (*R*) and its equivalent quaternion (*q*) where roll equals 45°, pitch equals 30°, and yaw equals 5°.

$$R = \begin{pmatrix} 0.8627 & 0.3002 & 0.4069 \\ 0.0872 & 0.7044 & -0.7044 \\ -0.4981 & 0.6432 & 0.5816 \end{pmatrix}$$

$$q = \left(0.8872 + 0.3797\mathbf{i} + 0.255\mathbf{j} - 0.06\mathbf{k}\right)$$

The only major drawback to using quaternions is their lack of intuitive understanding. For example, if a pilot says he was in a right turn with 15° angle of bank, 45° nose up, and in coordinated flight (yaw equals zero or no side-slip), then one can intuitively imagine the orientation of the aircraft based on the pilot's description. However, most people would not be able to even hazard a logical guess as to the orientation of the aircraft if the pilot said he positioned the aircraft using the equivalent quaternion,

$$q = \left(0.916 + 0.1206\mathbf{i} + 0.3794\mathbf{j} - 0.015\mathbf{k}\right).$$

THIS PAGE INTENTIONALLY LEFT BLANK

IV. EXPERIMENTATION METHODS

For many engineering applications, simulation is preferred over field experimentation due to the high cost typically associated with field experiments. However, Snowflake has been one exception to this generality. Artificial simulation of Snowflake and the air column is an extremely difficult task because there are so many factors at play. For instance, small changes in parachute rigging angles due to knots slipping in flight and minor misalignments of control lines are random occurrences which simply cannot be duplicated accurately in simulation. Additionally, wind is chaotic and non-linear. It is certainly possible to create an artificial wind, but the likelihood is not good that the simulation will be indicative of actual field conditions for the time and location of testing. Therefore, the majority of Snowflake's body of work went into actual field experimentation. The two experimental phases of Snowflake's life cycle are described in this chapter.

A. BENCH TESTS

Bench Tests were conducted in a laboratory on the campus of the Naval Postgraduate School in order to better understand the inner-workings of the Snowflake components. Specifically, an exorbitant amount of time went into testing the X-Monkey autopilot orientation and how the device records data. Furthermore, extensive analysis went into the study of how the X-Monkey represented its orientation with respect to an input of Euler angles (roll, pitch and yaw) and the respective quaternion outputs. Of note the, X-Monkey autopilot was designed to be flown lying flat in a quadcopter or radio-controlled (R/C) airplane. Due to the physical constraints of a Snowflake, the X-Monkey board cannot fly in this position, so it is mounted sideways in the vertical plane. Figure 15 depicts the two different X-Monkey orientations described.

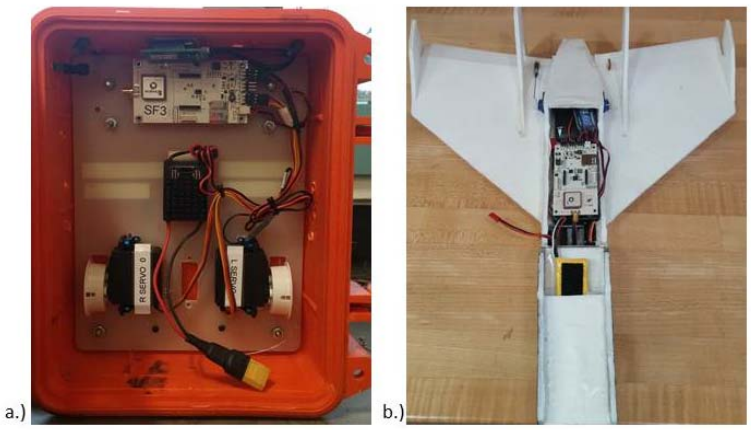

Figure 15. X-Monkey Orientations: a.) Mounted in a Snowflake and b.) Mounted in an airplane.

So a rotation from the X-Monkey to Snowflake frame of reference must be made. This equates to a rotation of 90° about both the yaw and pitch axis. The quaternion representing this rotation is q_{sf} = [0.5 0.5 0.5 -0.5].

In order to ensure the proper rotations were made, a MATLAB script was created and rudimentary tests were performed by turning the Snowflake in different directions to ensure rotation about all three axes were properly recorded. The MATLAB script also helped to compare the X-Monkey IMU sensed headings with known cardinal heading directions. The X-Monkey software from the developer comes with a non-intuitive IMU and accelerometer calibration tool, which often failed or captured invalid information. Thus, it was critically important to ensure the X-Monkey's heading matched expected headings prior to field testing evolutions.

B. MCMILLAN AIRFIELD TESTS

McMillan Airfield is a 1,067-meter runway tucked into a remote area of the Camp Roberts California National Guard Base, located approximately 25 kilometers north of Paso Robles, CA. The airfield is maintained and operated by a Naval Postgraduate School–funded organization known as the Center for Interdisciplinary Remotely-Piloted Aircraft Studies (CIRPAS), whose mission is to

“provide a base of operations for UAV flight activities for internal and customer aircraft from the military, scientific, and developmental arena” [15]. Figure 16 depicts the airspace surrounding Camp Roberts and the McMillan Airfield layout. For reference, the runway at McMillan Airfield is approximately aligned with headings 100 and 280. Also, the airspace immediately above McMillan Airfield is classified as Restricted Airspace. The Federal Aviation Administration (FAA) cautions that “restricted areas are established to separate activities considered to be hazardous to other aircraft” [16]. Prior coordination with the controlling authority for the restricted airspace and a special flight clearance from Air Traffic Control (ATC) is required prior to an aircraft entering a restricted area. As such, the restricted area above McMillan Airfield “can be activated by CIRPAS in coordination with the appropriate base enabling UAV flights without interference with, or to, other aircraft” [15]. Additionally, the restricted area is “surrounded by a series of Military Operating Areas (MOAs) to help facilitate Air Operations” [15].

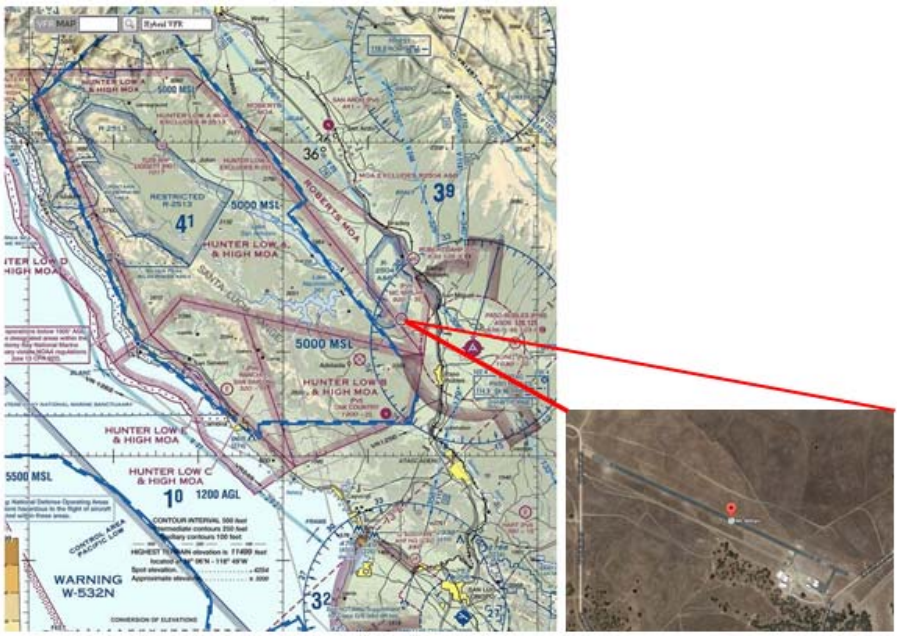

Figure 16. Visual Flight Rules Sectional Chart of the Camp Roberts Airfield (left) and an Aerial Photo of the McMillan Airfield (right). Source: [17], [18].

The topography surrounding McMillan Airfield gives rise to many different meteorological anomalies. The runway essentially sits in a valley with small foothills categorized by a slowly increasing gradient approximately 100 meters north of the runway and a small mountain chain with rapidly rising terrain about two kilometers to the south and west of the airfield. As one might surmise, the interaction between the wind and these topographical features can play havoc on wind estimation and small unmanned vehicle operations. The chaotic effects of the wind and topography interactions is typically in the form of rapidly shifting winds at different altitudes below 1,000 meters, updrafts caused by winds being forced up from the rising terrain, and thermal upwelling resulting from temperature deviations at the different altitudes. Therefore, the protected airspace and unique meteorological environments make McMillan Airfield an ideal location for testing and evaluating wind estimation methods.

V. WIND ESTIMATION

In aviation, winds are a meteorological phenomenon that must always be studied, understood, planned for, and reacted to. In extreme cases, flying against a severe headwind may lead to fuel starvation prohibiting an aircraft from reaching its destination, while a strong tailwind could produce excess airspeed (energy) during landing resulting in the aircraft departing the runway. Unlike an unmanned vehicle, "a human pilot learns to estimate the exposure to wind from experience, visual information, and 'seat-of-the-pants' feel. The human pilot predicts its effects and will use appropriate anticipatory guidance commands when changing course or maneuvering" [19]. In order for an unmanned vehicle to perform human-like tasks, algorithms must be programmed into its autopilot in order to provide some similar wind estimation abilities. This chapter discusses the techniques and tools that could be used to estimate winds in the interests of aerial delivery systems.

A. WIND ESTIMATION TECHNIQUES

"Although guidance of all aircraft is affected by the atmospheric motion relative to the Earth, that is, wind, micro-UAV's are especially susceptible. Localized wind field estimation, especially winds at low velocity, is difficult" [20]. As [20] describes, the aerodynamics associated with the ram-air parachute and the light payloads tested on Snowflake make it incredibly vulnerable to winds. The air column that the Snowflake descends through has constantly changing winds that vary in both magnitude and direction. Individual wind changes can be represented by a three-dimensional vector comprised of an x and y component (North and East) and a z component (altitude). Winds which occur in the vertical plane, commonly referred to as updrafts or downdrafts, are typically the product of rapidly changing temperature gradients in the air column, or the vertical wind occurs as a byproduct of the interaction between a horizontal wind and a vertical structure, known as mechanical wind.

The best way to obtain wind estimates is to measure them directly using a weather balloon or a miniature dropsonde (Figure 17) deployed right before the cargo airdrop. However, dropping an airsonde may not be practical; therefore other approaches must be explored in order to estimate the prevailing winds at least at the current altitude and above the ADS. Vertical winds are less prevalent in the atmosphere, thus their effects may be assumed to be zero in both estimation methods presented. This section reviews the theory and mathematical basis for two wind estimation methods.

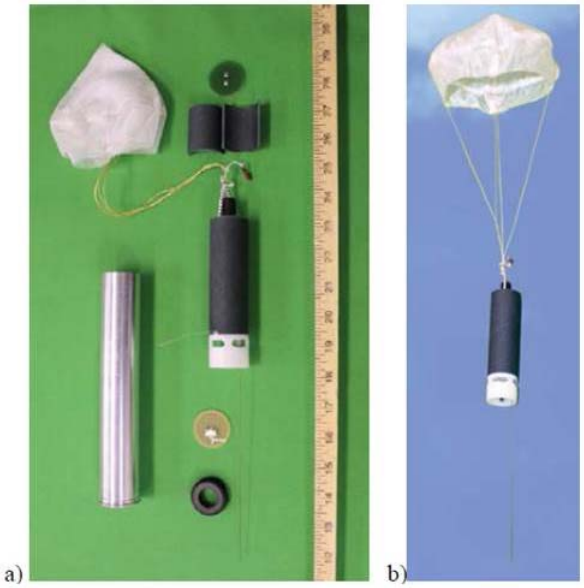

Figure 17. a) MAXMS Dropsonde components and b) MAXMS Dropsonde in flight. Source: [5].

1. The 360° Turn Method

The first method developed for Snowflake data testing and GNC algorithm incorporation is the 360° Turn method. Based loosely on Yakimenko's derivation in [5], a wind estimation method can be created "while executing a constant-control-input turn. Under no wind conditions [this] produces a right circle; in a windy environment it results in a stretched cycloid." Thus, the winds can be estimated using GPS data alone. As shown in Figure 18, in the absence of wind,

a constant turn will result in a 3D spiral. However, when a wind is present, this spiral gets elongated in the direction of the wind.

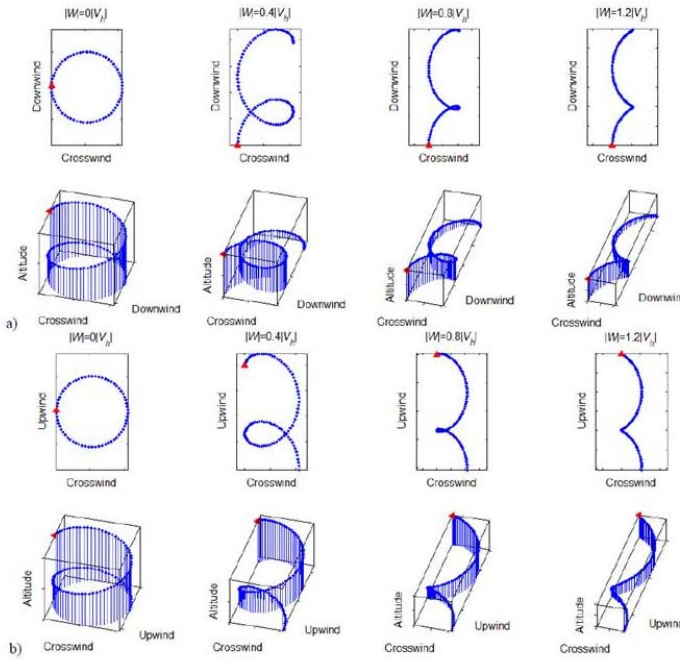

Figure 18. PADS Execution of 360° Turns when the Maneuver Starts in the a.) Downwind and b.) Upwind Direction. Source: [5].

Based on this theory, the algorithm notes Snowflake's GPS position, measured in degrees of latitude and longitude, each time the Snowflake passes through 360° of heading change. Then by examining the relationship between two consecutive 360° turn points, a flight trajectory can be estimated. The resultant difference between the flight trajectories of the two points is the effect of wind. From a structural design standpoint, this method is effective because the only sensor it uses is the GPS whose cost and weight have already been accounted for since a GPS card is already incorporated on the X-Monkey autopilot. Additionally, the requirement for a constant turn is not too far-fetched because many GNC algorithms will use an orbit, multiple turns over a fixed point, to descend from high altitudes and to prepare the PADS for landing. An example of the 360° Turn Method will be explored further in Chapter VI.

2. Multi-Sensor Method

The second wind estimation method incorporates a larger sensor suite in the hope of increasing the fidelity of the wind estimation algorithm. As with the 360° Turn Method, the Multi-Sensor Method utilizes GPS information, but only the GPS ground speed (V_{GPS}) and the GPS heading (ψ_{gps}). Additionally, the Multi-Sensor Method incorporates an Inertial Measurement Unit (IMU) to obtain the Snowflake heading (ψ_{SF}) and an estimate of an airspeed (V_{SF}). The foundation of the Multi-Sensor Method's algorithm as presented in [5] is as follows:

$$w_x = V_x - V_h \cos(\psi)$$
$$w_x = V_y - V_h \sin(\psi)$$

Here V_x and V_y are the respective directional components of the vehicle measured velocity from the GPS and V_h is the vehicle's airspeed measurement from the pitot tube. Adapting kinematic equations requires the Snowflake's GPS ground speed to be broken down into directional components. Combining this step and substituting Snowflake's parameters into the kinematic equations results in the following:

$$w_x = V_{x,GPS} \cos(\psi_{GPS}) - V_{SF} \cos(\psi_{SF})$$
$$w_y = V_{y,GPS} \sin(\psi_{GPS}) - V_{SF} \sin(\psi_{SF})$$

These two equations represent the winds sensed by Snowflake at a particular instance in time. In order to prove useful, the two component wind directions are combined into a magnitude and direction using the following equation:

$$\|W(t)\| = \sqrt{w_x^2 + w_y^2}$$
$$\angle W(t) = \tan^{-1}\left(\frac{w_y}{w_x}\right)$$

Consequently, the ADS now has an estimated wind magnitude and direction for every time instance *t*.

When compared to the 360° Turn Method, there is an increased cost for the Multi-Sensor Method in both financial and weight aspects due to the addition of new sensor hardware. However, the autopilots described in Chapter II, Pixhawk and X-Monkey, both come with IMUs already installed and integrated into their accompanying software packages.

B. POST-PROCESSING USING MATLAB

"Millions of engineers and scientists worldwide use MATLAB to analyze and design the systems and products transforming our world" [21]. Snowflake uses the software's extensive tools in order to post-process each flight's data recording. Onboard X-Monkey, flight data is recorded continuously from the instant power is applied to the autopilot's circuit board until the moment the power is removed. Consequently, the amount of data can reach astronomical sizes as a result of takeoff delays, flight duration, and the time required to recover the physical system and disconnect the battery. In order to facilitate and expedite the data processing effort, the X-Monkey data is automatically truncated into 7.2 megabyte packets. Using X-Monkey's proprietary software, the data packets are then converted to *.csv* files for manipulation in MATLAB. A typical SF flight from T-20 takeoff through Snowflake landing takes up two .csv data files. Each data file is made up of a maximum of 38,000 entries of 65 different flight parameters. This means that each *.csv* file can contain almost 2.5 million individual values! Although this is a large number, MATLAB is capable of handling this size file quite efficiently. Additionally, MATLAB can be used to create figures, or plots, to help users visualize data trends and explain their calculations. To increase the product's utility, there are a wide variety of add-on packages, or toolboxes, for a multitude of engineering applications. MATLAB is the go to software for the post processing of Snowflake missions due to its ability to handle large data files and produce meaningful figures. The core MATLAB script used to process the Snowflake data and create the two wind estimation algorithms is included as an Appendix.

THIS PAGE INTENTIONALLY LEFT BLANK

VI. WIND ESTIMATION RESULTS

This chapter presents the figures created by the Snowflake's post-processing MATLAB scripts along with accompanying description, analysis, and additional comments pertaining to the figures. Of note, when comparing the 360° Turn and Multi-Sensor Methods, the resultant wind estimations are only as reliable as the equipment used. This may seem like an obvious observation, but a repeated diagnostic review of component functionality is required due to the inability of some system components to provide accurate measurements after numerous impacts with the ground. For comparison, the results of both wind estimation methods presented in Chapter V are products of a dataset created in the restricted airspace above McMillan Airfield during a single Snowflake test flight on July 21, 2016.

A. TEST RESULTS OVERVIEW

This section portrays commonalities from the post-processing of the Snowflake data. Broad trend analysis can be conducted and hypothesis can be gleaned from these general outcomes. Figure 19 depicts a top-down view of the Snowflake's flight profile. The red square represents the point at which the ram-air canopy deployed and the green diamond is the impact point of the Snowflake, while the blue dots connecting the two shapes are the flightpath in two dimensions.

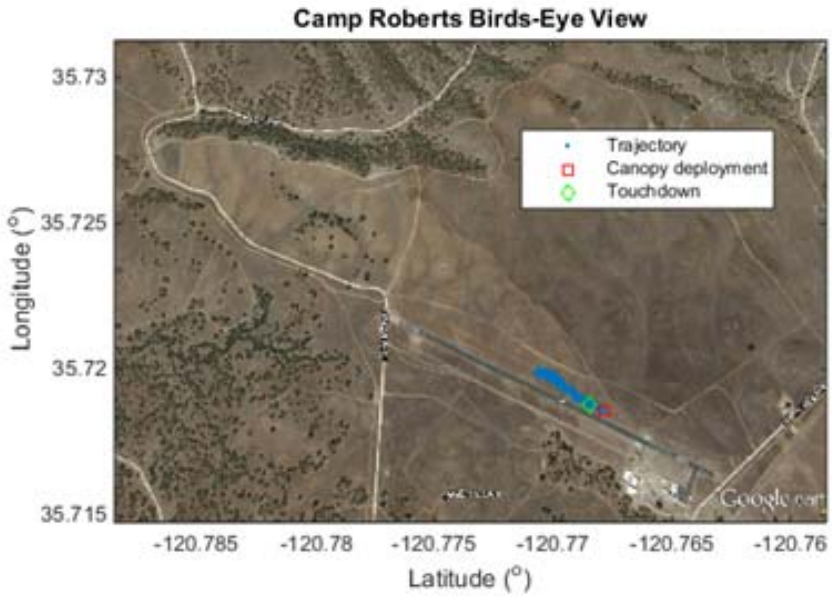

Figure 19. Snowflake Flight Path at McMillan Airfield on July 21, 2016. Adapted from [18].

From this figure, one can see that Snowflake initially drifted along the runway from right to left; and then it reversed course and landed at almost the exact point below where the canopy opened. However, this figure lacks a lot of detail especially since the Snowflake appears to have basically flown along the same path for much of the flight time. For added help with flightpath visualization, Figure 20 depicts the 3-dimensional (3D) representation of the same flight. This illustration uses the same red square and green diamond symbols to show start and stop points for the flight, but this figure also provides insight into the flight path with regards to altitude measured in meters AGL. From Figure 20, one can clearly make out the spiral shape created as the Snowflake performs consecutive, descending 360° turns.

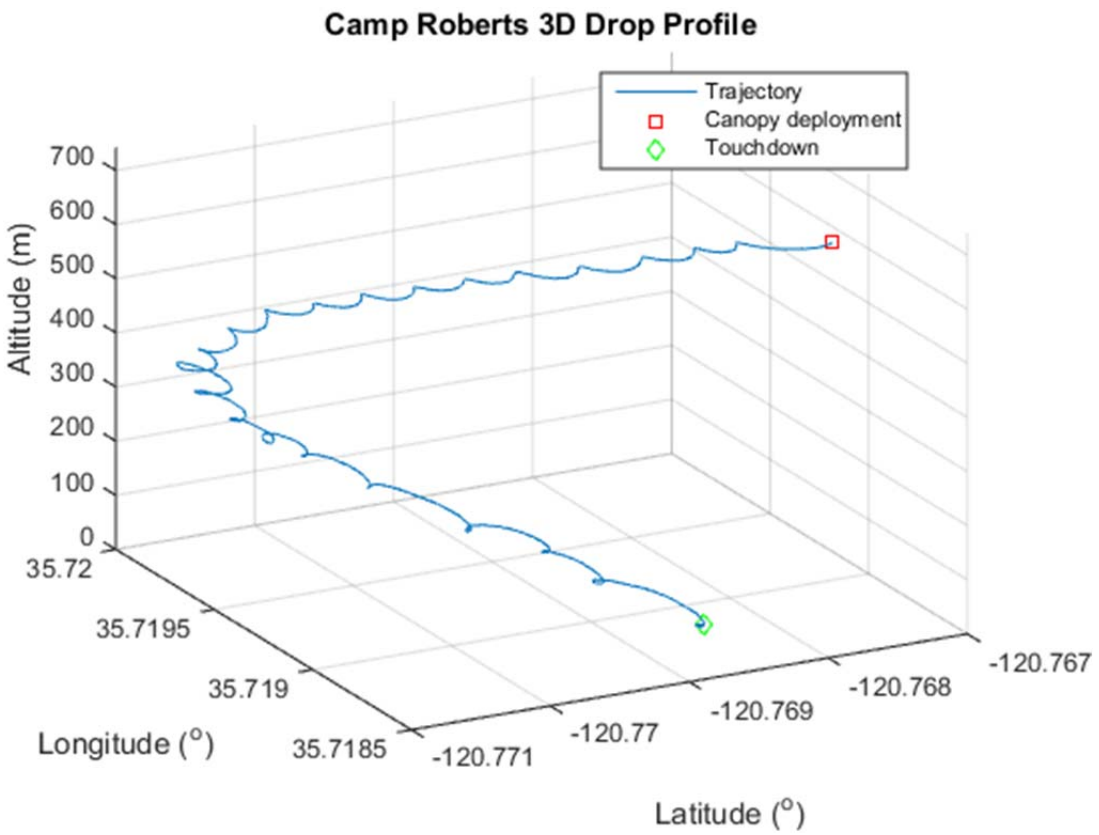

Figure 20. 3D Representation of the Snowflake Flightpath on July 21, 2016.

For an object released from an aircraft following a truly ballistic flight, the aircraft's velocity and orientation play a major factor [8]. However, after examining the rapid decay of ground speed from more than 30 meters per second to seven meters per second in Figure 21, one can ascertain that the near instantaneous deployment of Snowflake's parachute negates the majority of the aircraft's velocity and orientation effects. Upon closer inspection, this airspeed decrease can be seen on the far left of the ground speed plot of Figure 21.

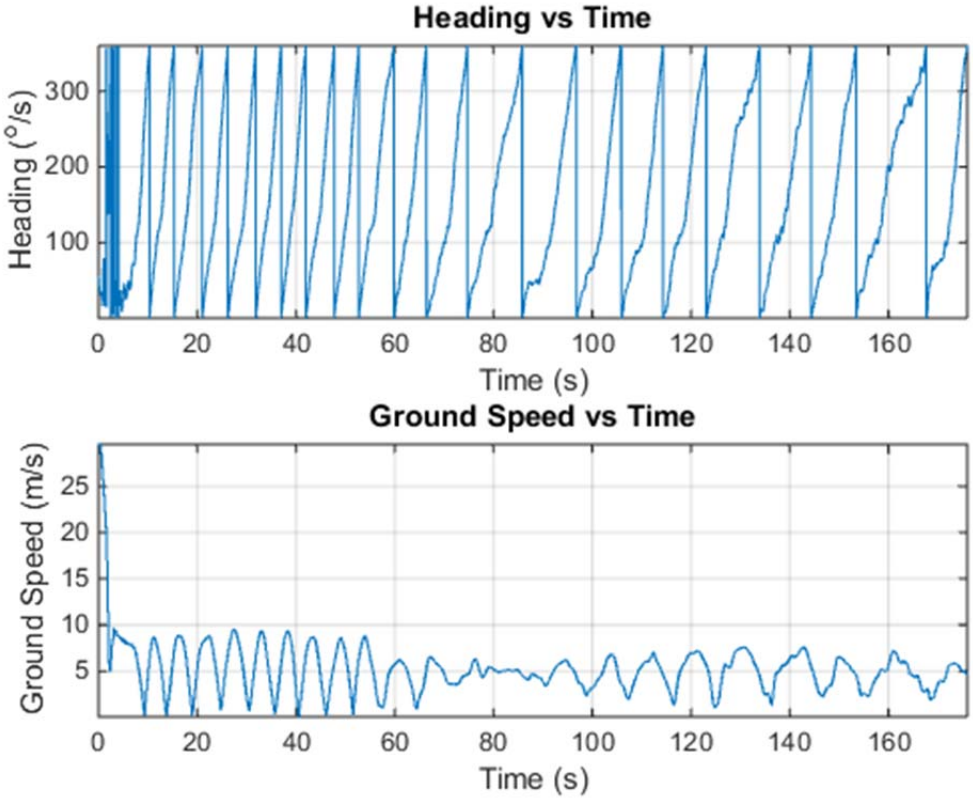

Figure 21. Rapidly Decaying Effects of UAV Airspeed as Seen by Snowflake.

The next figure that the wind estimation MATLAB script produces is actually three plots in one, which helps the analyst understand the roll, pitch, and yaw, (sometimes referred to as phi, theta, and psi) relationships. Note the discontinuities, represented as vertical lines in the yaw plot. This is not erroneous data, but simply how MATLAB processes the Snowflake passing from 360° to 001°. In general, the pitch and roll plots can be used to determine overall system stability during the flight. In Figure 22, the roll and pitch show some oscillations, but their mean is fairly constant. Adding additional weight to the payload or a spreader device to the suspension lines may help decrease some of these oscillations.

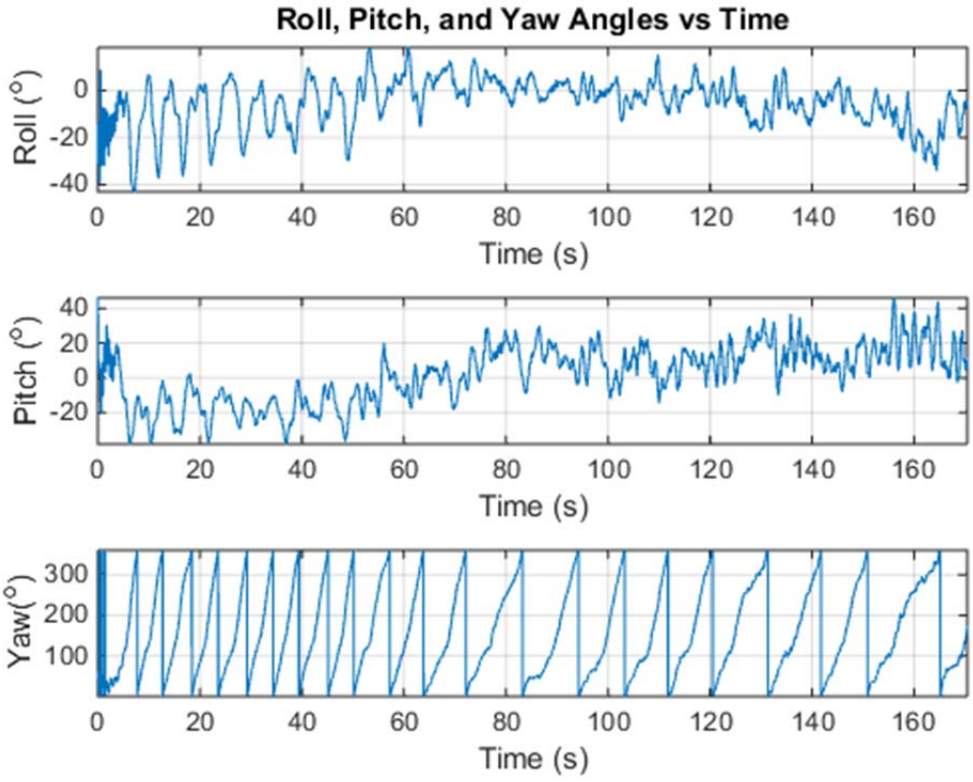

Figure 22. Time History of Roll, Pitch, and Yaw Angles.

In a similar plot, Figure 23 shows the roll, pitch, and yaw rates as a function of time. The data is quite noisy because there is a lot of small motion occurring during the Snowflake's descent. To aid in the analysis, a red dashed line has been added to denote the average values for the corresponding axis' rate of motion.

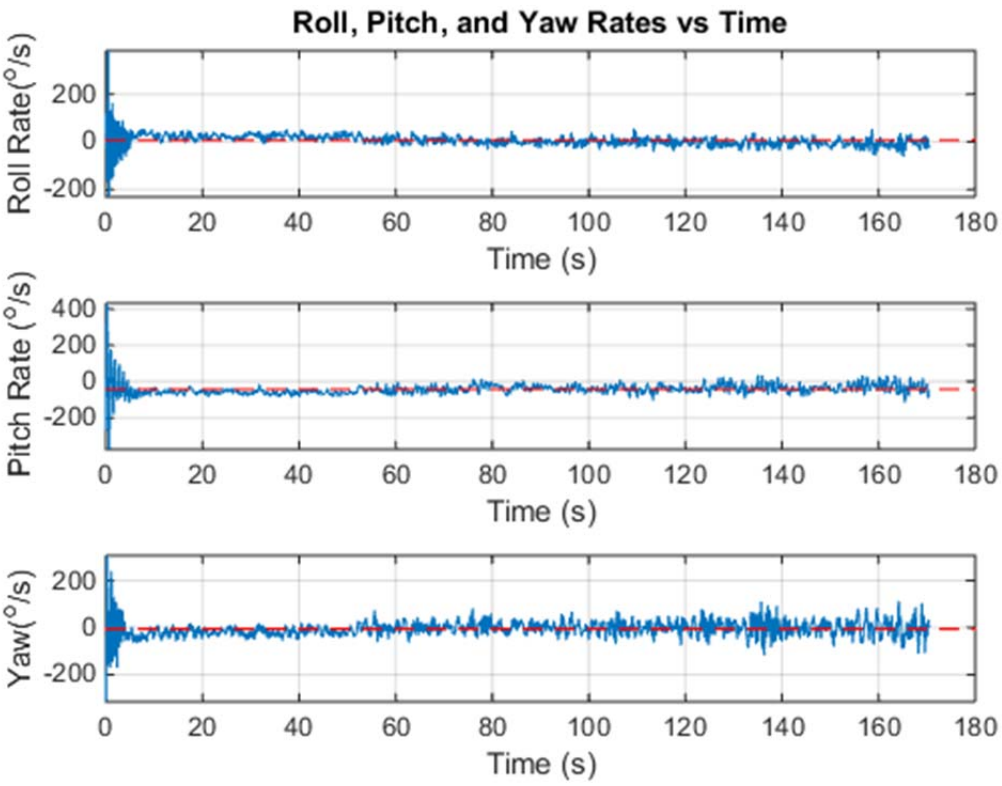

Figure 23. Time History of Roll, Pitch, and Yaw Rates.

Another tool used during the analysis is Figure 24, which represents the Snowflake's yaw angle, often called heading, and ground speed as a function of time. Notice the correlation between heading and ground speed maxima. If there is a prevailing wind present, then the ground speed maxima should occur at the same heading each time the Snowflake circles around. In other words at the corresponding heading of the maxima locations, the Snowflake is encountering the maximum tailwind. Subtracting 180° from this heading will yield the approximate wind direction. In Figure 24, by visual inspection alone one can infer that there is a wind present from 100° for about the first 80 seconds, and then the wind shifts to an approximate heading of 290. In aviation it is customary to refer to winds as "from" a certain direction, so a westerly blowing wind is said to be blowing *from* 090. For example, if a balloon was released it would drift due west

because it was impacted by a wind *from* 090. For the remainder of this analysis, winds will be assumed *from* a direction unless otherwise stated.

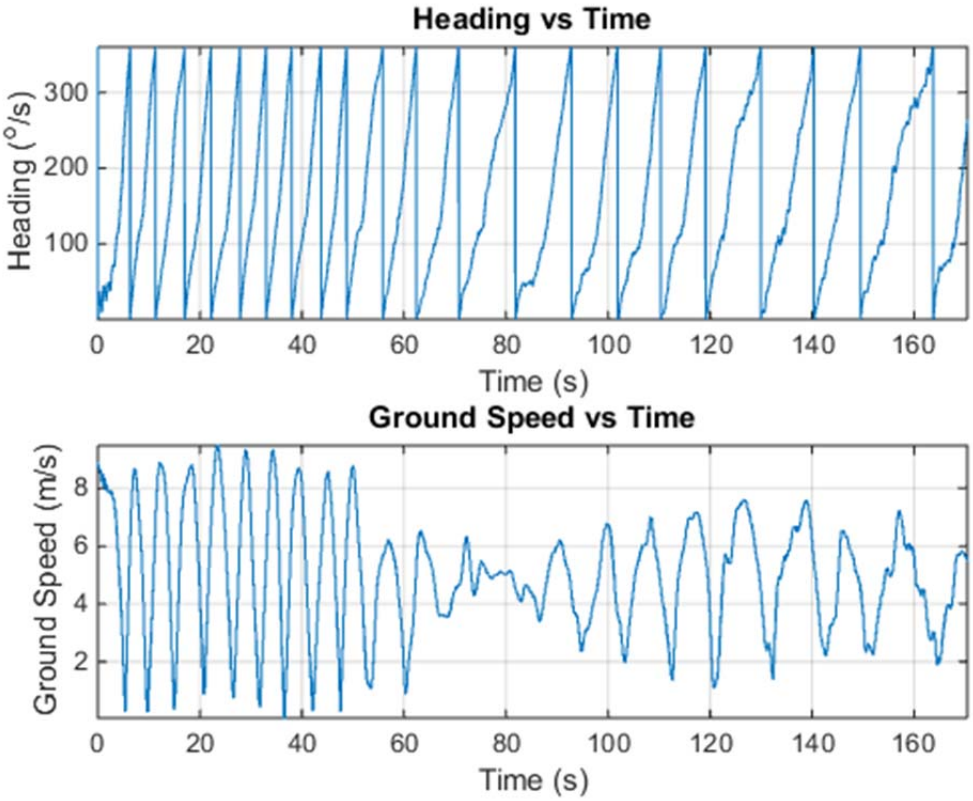

Figure 24. Heading and Ground Speed Correlation.

Finally, before employing multiple sensors, it is imperative that the user verify that the GPS and IMU are supplying similar heading indications. Completing this task ensures the accuracy of the figures generated by the 360° Turn Method, which uses GPS exclusively, and the Multi-Sensor Method, which incorporates IMU and GPS data. Figure 25 depicts the heading values for the IMU, in blue, and the GPS, in red. The figure shows the heading values are very similar except the GPS lags the IMU. The latency in GPS data is a product of sample rate. The GPS acquires data at 5 Hz which equates to 5 position recordings per second while the IMU records at a speed of 100 Hz resulting in

100 readings per second. The difference between the two sampling rates produces the lag effect shown by the GPS in Figure 25.

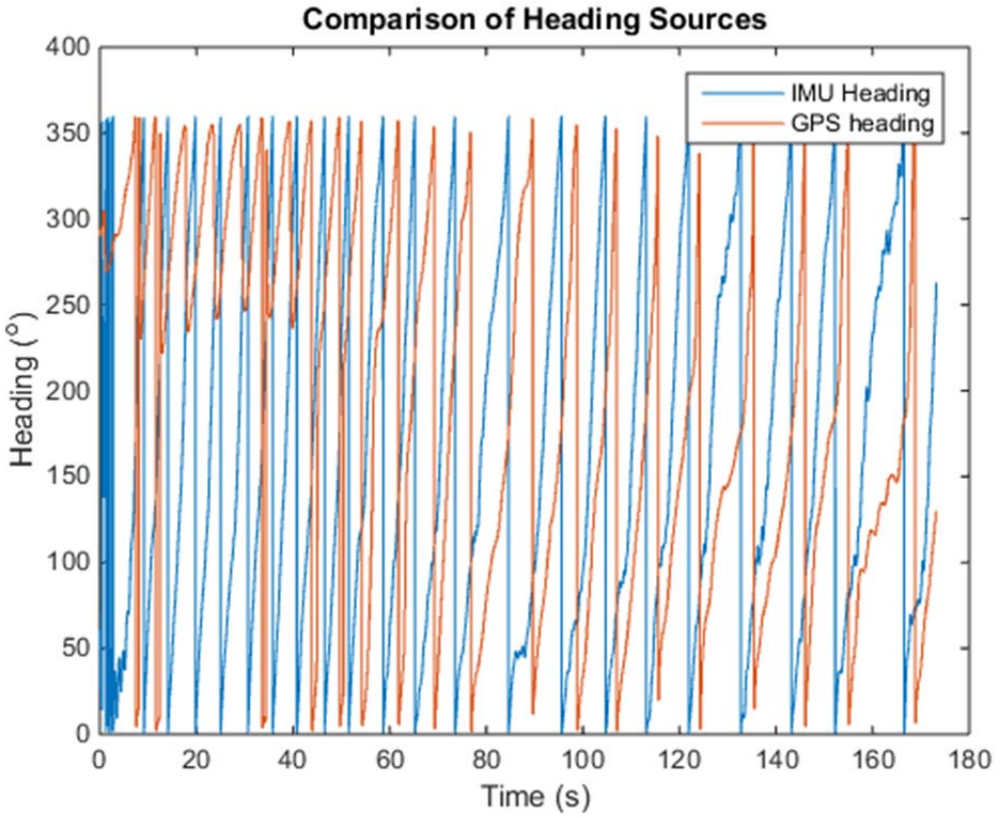

Figure 25. Comparison of Heading Sources.

B. 360° TURN METHOD RESULTS

In order to test the 360° Turn Method, a constant turn was commanded by the X-Monkey autopilot. This test was not trying to accurately land on a specific target, rather it was used to gather wind data to test the two wind estimation methods for application to subsequent flights. Through post-processing calculations, it was noted that Snowflake descended while turning at approximately 47° per second, completing a full 360° turn about once every 7.7 seconds. From the plot on the right side of Figure 26, one can see that at high altitudes, greater than about 425 meters, the Snowflake was influenced by a wind from the east, approximately 110°. Then, for about 100 meters of descent, the

winds basically dropped off to a very small value, less than 1 meter per second as seen on the left plot of Figure 26. Finally, the winds picked back up, but this time out of the west from approximately 290°.

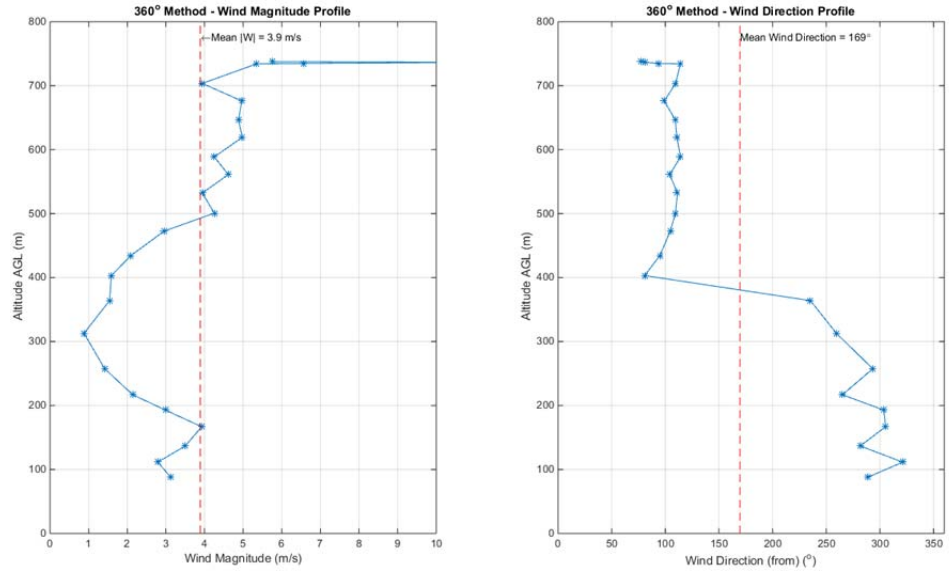

Figure 26. 360° Turn Method Estimate.

This method, using GPS alone, gives some valuable insight into the wind conditions aloft. It also clearly illustrates what an egregious mistake it is to try and use a constant wind value for all altitudes or no wind data at all. Additionally, this data is extremely valuable for planning future drops which are striving for accuracy.

At first glance the 180° wind direction change looks a little suspect. However, when compared to the flight path and 3D profile of Figures 19 and 20, the 360° Turn Method wind directions seem logical. Specifically, Figure 19 depicts the Snowflake initially flying from right to left on a heading approximately equal to the runway heading (280°). Then Figure 20 corroborates the drop in winds by showing a relatively straight descent between 425 and 325 meters. Finally, Figure 19 echoes the wind shift to winds from the west as the Snowflake's path heads east. Thus, the 360° Turn Method is capable of

predicting wind velocity and direction for a PADS when a constant turn is induced.

C. MULTI-SENSOR METHOD RESULTS

The Multi-Sensor Method has significantly more data points than the 360° Turn Method. For instance, in the data from the July 21, 2016, flight test, the 360° Turn Method consisted of only 22 data points because the Snowflake made 22 turns of 360° during its three-minute flight. In comparison, during the same flight, the Multi-Sensor Method captured 16,727 data points. Figure 27 depicts the wind magnitude and estimation for all these data points.

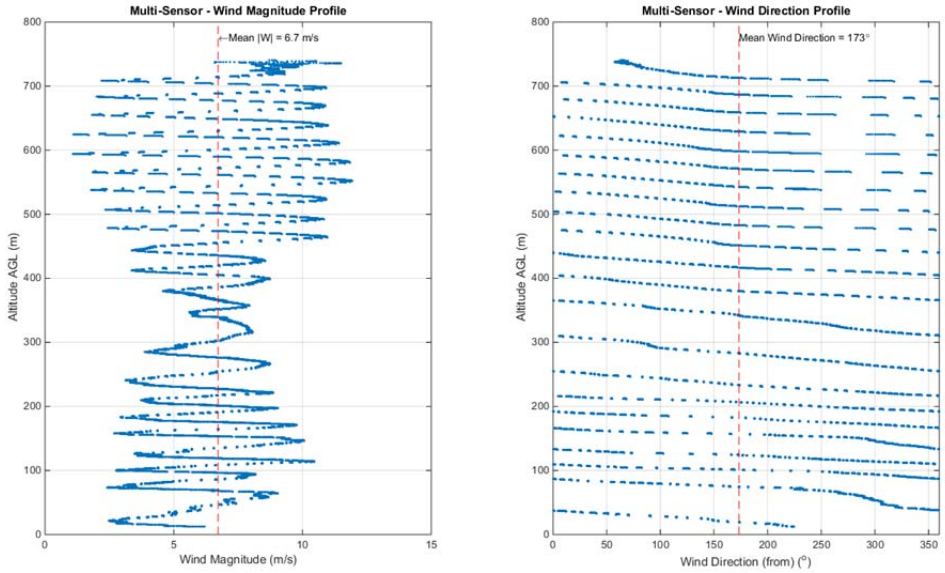

Figure 27. Multi-Sensor Method Wind Profiles with Repetitive GPS Data Estimate.

At first glance, the plots of Figure 27, especially the wind direction profile on the right, are very noisy and difficult to comprehend. Upon further inspection it was determined that the noise was due to repetitive GPS data recordings. As previously discussed, the GPS operates at a much slower frequency, capturing only five new readings per second. However, the IMU is capturing 100 new

readings during that same second, so when the X-Monkey records the data, it creates 20 repetitive entries of the GPS data to ensure there are 100 total values recorded for both the GPS and IMU. In order to cut out the repetitive data, the *unique()* function in MATLAB was used to isolate the time instance when new GPS data was recorded. As a result, the 16,727 data points were reduced by almost 95% and trimmed down to 845 distinct data points. Figure 28 depicts the same categorical information as Figure 27, but with only the 845 data points included.

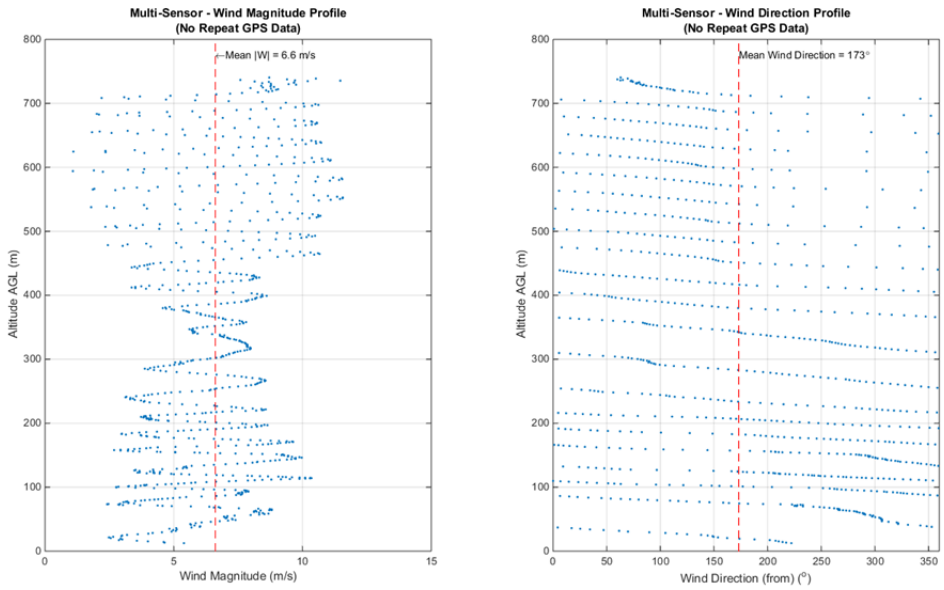

Figure 28. Multi-Sensor Method Wind Profiles without GPS Repetition.

The left plot, of what appears as a zig-zagging wind magnitude, clearly shows a headwind and tailwind presence as the Snowflake descends through altitude. The wind magnitude decays in magnitude around 400 meters and then increases again after about 100 meters of descent; all of which is consistent with the review of previously discussed method. Unfortunately, even with a 95% reduction in data points, the wind direction profile looks like a bunch of non-coherent zig-zags. However, upon closer inspection one can see there are data

point concentration areas in the upper left and lower right quadrants of the plot. This follows the wind estimation of the 360° Turn Method, but requires a little deeper inspection to ensure both methods align. A good tool for examining the areas of data concentrations is the histogram provided as Figure 29.

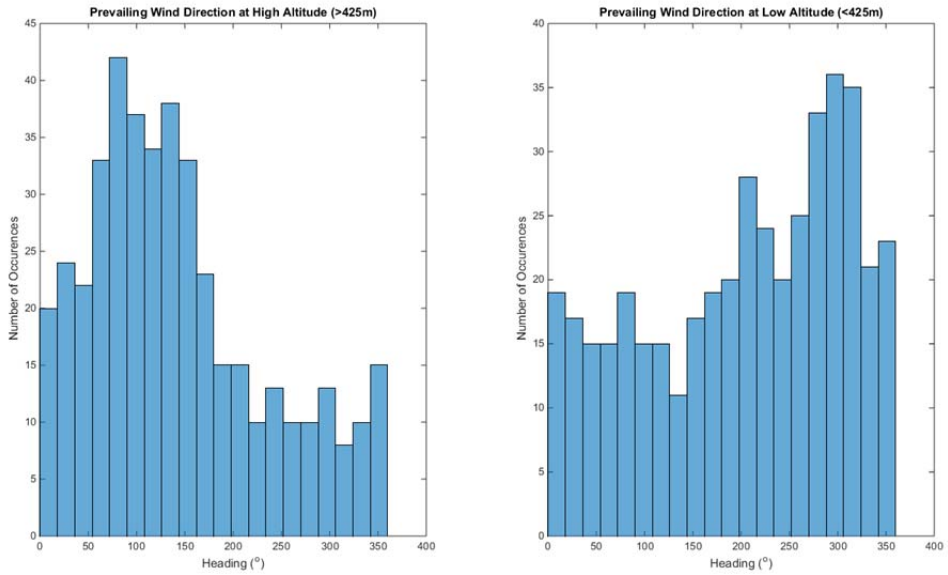

Figure 29. Histogram of Wind Directions.

The histogram shows the frequency of data points whose values were within certain bounds. From Figure 30, one can see that at altitudes greater than 425 meters, the winds were out of the east, south east between 095° and 145° and below 425 meters the winds were dominant in the region between 275° and 325°. Again, this estimation is very similar to the results of the 360° Turn Method and directly correlates to the flight profiles of Figures 19 and 20.

D. USING WIND ESTIMATES

It should be noted that no matter how accurate the method for estimating winds, the estimate only provides information of the wind column above the current altitude. In other words, there are no data points available to describe the winds below the current altitude of the PADS (see [22] for a wind estimation

technique of underlying altitudes using unscented Kalman filters). As such, the presented wind estimates should be used with caution.

To illustrate this principle, Figure 30 shows the same bird's eye view used in Figure 19, but this time incorporating the predicted touchdown points based on a blind use of the wind estimates provided by the 360° Turn Method and Multi-Sensor Method. The markings on the figure follow the assumption that wind estimate obtained for the current altitude remains unenhanced all the way to the ground.

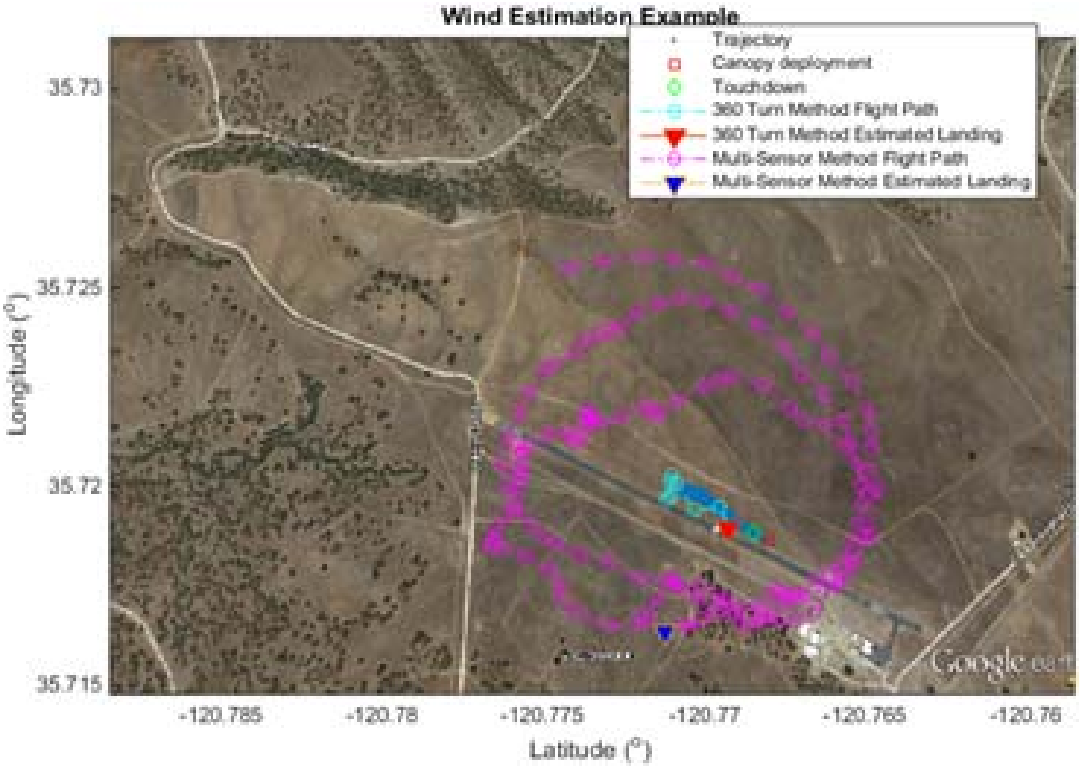

Figure 30. Wind Estimation Example. Adapted from [18].

Figure 31 shows a zoomed-in perspective of the same flight path and estimated landing points. For the 360° Turn Method, the estimated touchdown point locations, depicted with cyan circles, start at 300 meters. This is because for most PADS, it is the last 300–100 meters that is the most critical for wind analysis (this is where the decision is made to exit the energy management

maneuver and proceed to the target [5]). The solid red triangle denotes the last predicted landing point based on the wind estimate at about 90 meters AGL (see Figure 26). This final touchdown point is located approximately 75 meters circular error probable from the actual landing point.

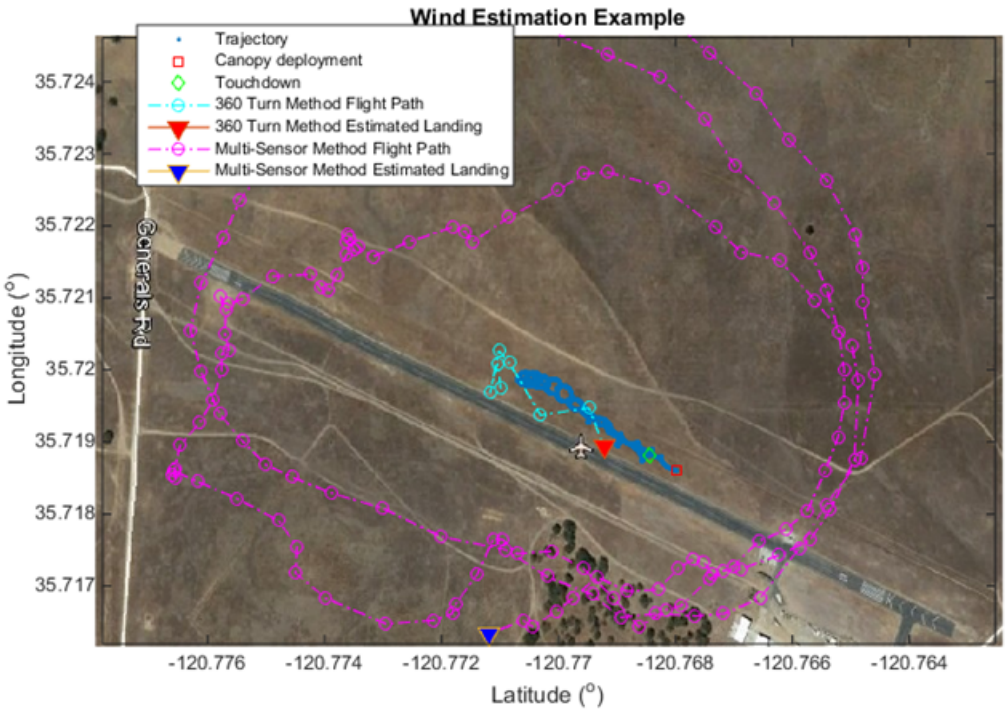

Figure 31. Zoomed-In Wind Estimation Example. Adapted from [18].

When JPADS began over 20 years ago, the Department of Defense (DOD) defined an accurate PADS landing as one which lands within 100 meters of its intended target [5]. Using this metric as a measuring stick, the 360° Turn Method seems capable of providing wind estimates that meet or exceed the DOD requirements.

The touchdown points, predicted by the Multi-Sensor Method's wind estimate, are denoted by the magenta circles, and strongly follow the semi-incoherent wind magnitude and direction data depicted in Figure 28. For this method, only the last 100 meters of the descent are depicted because including additional points from above 100 meters saturates the figure with an exorbitant

amount of data. The cyclonic nature of the Multi-Sensor Method's landing point estimates are caused by the pitch and roll oscillations as described in Chapter VI, Section A (see Figures 22 and 23). As a result of the variability, its last estimated landing point based on the previous touchdown estimate of the wind and denoted as a solid blue triangle, happens to be more than 350 meters away from the actual Snowflake landing point.

Both wind estimation methods provide meaningful insight into the effects of wind on an ADS. For many engineering applications, more data points are better. However, for the wind estimation methods provided in this chapter, the 360° Turn Method developed a better wind estimation solution with fewer data points. Fewer data points also requires less computing power and less course corrections, which is important to keep in mind because it may be incredibly difficult to produce a stable GNC algorithm if it is constantly inundated with course corrections as a result of the multitude of data points delivered by the Multi-Sensor Method.

THIS PAGE INTENTIONALLY LEFT BLANK

VII. CONCLUSION

Autonomous vehicle interest has seen an exponential growth trend over the past few decades. As a result, hobbyists and civilian companies are rapidly producing affordable sensor and autopilot components with increased capabilities. This thesis presents two wind estimation methods designed to complement the increased capabilities of the commercial off-the-shelf sensors and autopilots. The 360° Turn Method provides an outstanding tool for wind estimation in orbits due to its dependency on consecutive turns. The financial burden to incorporate the 360° Turn Method is minimal since most commercial off-the-shelf autopilots have a GPS sensor incorporated in their basic configuration. In comparison, the Multi-Senor Method poses increased flight profile flexibility to the user because it is not restricted only to orbiting profiles. However, the increased flexibility of the Multi-Sensor Method does come at a both a financial and total system weight cost due to the addition of the pitot tube system. Additionally, the results of the Multi-Sensor Method include a lot more variability, or noise, due to the rapid IMU sampling rates. Similarly, the plethora of data coming from the Multi-Senor Method may require additionally computationally intensive filtering processes in order to provide a wind estimation suitable for incorporation into a GNC algorithm.

Finally, it should be noted that the wind estimates of the current altitude still do not show the entire picture because the winds below the PADS are still unknown. The wind algorithms created for this thesis prove conclusively that commercially manufactured autopilots are capable of gathering reliable data for wind estimation. The extrapolated data is suitable for post-processing wind estimation and incorporation in a GNC algorithm to yield increased flight accuracy. Without the employment of wind estimation methods, a parachute aerial delivery system is subject to the chaotic effects of wind, spoiling any hopes of an accurate landing.

THIS PAGE INTENTIONALLY LEFT BLANK

APPENDIX

```
% Snowflake Wind Profile Generator
% LCDR O'Brian

clear all
close all
clc

%% Read in the data
% This section is written for 2 data files but can be adjusted. Typical
% Snowflake drops do not take up an entire file, but occasionally they
do
% overlap between two files depending on XMonkey data recording times.
CampRoberts = 1;
data_raw1 = csvread('201607211823.csv',1,0); %parsed csv file
data_raw2 = csvread('201607211829.csv',1,0);

data_raw = data_raw1;
data_raw(length(data_raw1)+1:length(data_raw1)+length(data_raw2),:) =
data_raw2;


data_raw(:,1) = [0:length(data_raw)-1];
%% Truncate the data
% At this time I am only interested in the drop profile. In other words
% from the SF release until it hits the ground. This segment of the
script
% prints a plot and allows the user to choose the release point and
% touchdown point.

%plots altitude at index number
plot((data_raw(:,1)-data_raw(1,1))+1,data_raw(:,22))
title({'Complete Flight Profile';'Choose Release Point and Point of
Impact'});
xlabel('Time of Flight (s)');
ylabel('Altitude (m)');
%selects points to collect data between
[idx,~] = ginput(2);
%rounds index number
idx = round(idx);
%trims tail of data
data_raw([idx(2):end],:) = [];
%trims head of data
data_raw([1:idx(1)],:) = [];

%% Create a new Table of the data w/ proper labels on the fields

n=length(data_raw);
Index = (data_raw(:,1)-data_raw(1,1))+1; %datapoint index
Time_s = data_raw(:,5)-data_raw(1,5); %zeros out time
Altitude_m = data_raw(:,22); %unadjusted raw altitude
```

```
Longitude_W = data_raw(:,21); %grid x position
Latitude_N = data_raw(:,20); %grid y position
X_Position_m = (Longitude_W(:,1)-Longitude_W(1,1))*111303; %converts
GPS longitude into meters with initial at zero
Y_Position_m = (Latitude_N(:,1)-Latitude_N(1,1))*110575; %converts GPS
latitude into meters with initial at zero
Roll_deg = wrapTo180((data_raw(:,28))); %roll in degs
Pitch_deg = data_raw(:,29); %pitch in degs
Yaw_deg = wrapTo360(data_raw(:,30)); %yaw in degs
Acc_x_ms2 = data_raw(:,32); %x acceleration in m/s2
Acc_y_ms2 = data_raw(:,33); %y acceleration in m/s2
Acc_z_ms2 = data_raw(:,34); %z acceleration in m/s2
Roll_Rate_degs = data_raw(:,35); %roll rate in deg/s
Pitch_Rate_degs = data_raw(:,36); %pitch rate in deg/s
Yaw_Rate_degs = data_raw(:,37); %yaw rate in deg/s
[~, ~, Rho_Air_Kgm3, ~] = atmosphere(Altitude_m); %computes air density
as a function of alitude
Rho_Air_Kgm3 = Rho_Air_Kgm3'; %reorients
Dynamic_Pressure_Pa = 999*9.81*0.00508.*data_raw(:,42)/1000; %dynamic
pressure reading in Pa
Air_Speed_ms = ((2*Dynamic_Pressure_Pa)./(Rho_Air_Kgm3(:,1))).^0.5;
%air speed in m/s
Temp_c = data_raw(:,50); %temperature in celcius
Static_Pressure_Pa = data_raw(:,51); %static pressure reading in Pa
PWM_0_out = data_raw(:,59)-1000; %servo out [0,1000]
Deflection_inches = ((PWM_0_out-1000*.94)/(0-1000*.94)*(9+9))-9;
Servo_Current_A = data_raw(:,43)/66.5; %Servo Current in Amps
User_1 = data_raw(:,6);
User_2 = data_raw(:,7);
User_3 = data_raw(:,8);
ground_speed = data_raw(:,23);
yaw_rate = data_raw(:,37);
GPS_Hdg = data_raw(:,24);

%organize data into a table
T =
table(Index,Time_s,Altitude_m,Longitude_W,Latitude_N,X_Position_m,Y_Pos
ition_m,Roll_deg,Pitch_deg,...

Yaw_deg,Acc_x_ms2,Acc_y_ms2,Acc_z_ms2,Roll_Rate_degs,Pitch_Rate_degs,Ya
w_Rate_degs,Temp_c,...

Air_Speed_ms,Dynamic_Pressure_Pa,Static_Pressure_Pa,PWM_0_out,Deflectio
n_inches,Servo_Current_A,...
 User_1,User_2,User_3,ground_speed,yaw_rate,GPS_Hdg);

UsefulData = table2array(T);

%% Create Birds-Eye View Plot
figure()
if (CampRoberts == 1)
 CRImage = imread('CPRobertsflip','jpeg');
 DZimage = CRImage(:,:,1:3);
 image([-120.788473 -120.758492],[35.714764 35.731265], DZimage)
 axis([-120.788473 -120.758492 35.714764 35.731265])
```

```
 set(gca,'YDir','normal')
 daspect([1 cosd(UsefulData(end,5)) 1])
 hold on
end
plot(UsefulData(:,4),UsefulData(:,5),'.')
plot(UsefulData(1,4),UsefulData(1,5),'rs')
plot(UsefulData(end,4),UsefulData(end,5),'gd')
title('Camp Roberts Birds-Eye View');
h=legend('Trajectory','Canopy deployment',...
 'Touchdown','location','best'); set(h,'fontsize',8);
xlabel('Latitude (^o)');
ylabel('Longitude (^o)')
axis tight

%% Create similar 3D Plot
figure()
plot3(UsefulData(:,4),UsefulData(:,5),UsefulData(:,3)-280,'-');
hold on
plot3(UsefulData(1,4),UsefulData(1,5),UsefulData(1,3)-280,'rs')
plot3(UsefulData(end,4),UsefulData(end,5),UsefulData(end,3)-280,'gd')
title('Camp Roberts 3D Drop Profile');
xlabel('Latitude (^o)');
ylabel('Longitude (^o)');
zlabel('Altitude (m)');
h=legend('Trajectory','Canopy deployment',...
 'Touchdown','location','best'); set(h,'fontsize',8);
zlim([0 inf]);
view(-28,27);
grid
hold off

%% Plot Roll, Pitch, Yaw
figure()
subplot(311)
plot(UsefulData(:,2),UsefulData(:,8)+90); % Roll Plot
hold on
title('Roll, Pitch, and Yaw Angles vs Time')
xlabel('Time (s)');
ylabel('Roll (^o)');
grid on
axis tight
hold off

subplot(312)
plot(UsefulData(:,2),UsefulData(:,9)); % Pitch Plot
hold on
xlabel('Time (s)');
ylabel('Pitch (^o)');
grid on
axis tight
hold off

subplot(313)
plot(UsefulData(:,2),UsefulData(:,10)); % Heading Plot
hold on
```

```
xlabel('Time (s)');
ylabel('Yaw(^o)');
grid on
axis tight
hold off

%% Plot Roll, Pitch, and Yaw Rates
% Compute average rates:
 rollrateAVE = mean(UsefulData(:,14));
 pitchrateAVE = mean(UsefulData(:,15));
 yawrateAVE = mean(UsefulData(:,16));


figure()
subplot(311)
plot(UsefulData(:,2),UsefulData(:,14)); % Roll Rate Plot
hold on
plot(get(gca,'xlim'),[rollrateAVE rollrateAVE],'r--');
title('Roll, Pitch, and Yaw Rates vs Time')
xlabel('Time (s)');
ylabel('Roll Rate(^o/s)');
grid on
axis tight
hold off

subplot(312)
plot(UsefulData(:,2),UsefulData(:,15)); % Pitch Rate Plot
hold on
plot(get(gca,'xlim'),[pitchrateAVE pitchrateAVE],'r--');
xlabel('Time (s)');
ylabel('Pitch Rate (^o/s)');
grid on
axis tight
hold off

subplot(313)
plot(UsefulData(:,2),UsefulData(:,16)); % Heading Rate Plot
hold on
plot(get(gca,'xlim'),[yawrateAVE yawrateAVE],'r--');
xlabel('Time (s)');
ylabel('Yaw(^o/s)');
grid on
axis tight
hold off

%% Plot Heading vs Ground Speed

figure();
 subplot(211)
 plot(UsefulData(:,2),UsefulData(:,10)); % Heading Plot
 hold on
 title('Heading vs Time')
 xlabel('Time (s)');
 ylabel('Heading (^o/s)');
```

```
 grid on
 axis tight
 hold off
 subplot(212)
 plot(UsefulData(:,2),UsefulData(:,27)); % Ground Speed Plot
 title('Ground Speed vs Time');
 ylabel('Ground Speed (m/s)');
 xlabel('Time (s)');
 grid on
 axis tight
 hold off

 %% Wind Estimation for entire flight:

 % Average Wind Magnitude:

 Vgs_max = max(UsefulData(:,27)); %Max Ground speed during drop
 Vgs_min = min(UsefulData(:,27)); %Min Ground speed during drop

 Vav = 0.5*(Vgs_max +Vgs_min); %Average Wind Magintude

 Wind = max(Vgs_max-Vav,Vgs_min+Vav);
 Wind_kts = Wind*1.94384; % Convert m/s to knots
 %fprintf('The estimated wind velocity is %0.3f w/s or about %0.1f kts\
n',Wind,Wind_kts);

%% Wind Estimation METHOD 1:

 fullturn = 0;
 turndata = diff(UsefulData(:,10));
 for m =1:n-1
 if abs(turndata(m)) > 355
 fullturn = fullturn +1;
 end
 end

DATA = zeros(fullturn,3);
mm=1;
for m = 1:n-1
 if abs(turndata(m,1)) > 300

 DATA(mm,1) = UsefulData(m,2); % Time
 DATA(mm,2) = UsefulData(m,5); % Lat
 DATA(mm,3) = UsefulData(m,4); % Lon
 DATA(mm,4) = UsefulData(m,3); % Alt
 mm=mm+1;
 end
end


for m = 1:length(DATA)
 if m < length(DATA)
 h = DATA(m,4); % altitude1
```

```
 h2 = DATA(m+1,4);
 SPHEROID = referenceEllipsoid('wgs84', 'm'); % Reference ellipsoid w/
units of 'm'
 [N, E] = geodetic2ned(DATA(m,2), DATA(m,3), h, DATA(m+1,2),
DATA(m+1,3), h2, SPHEROID);
 METHOD1(m,1) = DATA(m+1,1)—DATA(m,1); %Calculate flight time between
points
 METHOD1(m,2) = norm([N, E]); %Calculate distance between two points
(in m)
 METHOD1(m,3) = azimuth(DATA(m,2), DATA(m,3), DATA(m+1,2), DATA(m+1,3),
SPHEROID, 'degrees'); % Calculate Wind direction
 % This loop puts the wind in "aviation sense" where winds come from a
 % direction. Normally this would required adding or subtracting 180
 % degrees.
 if METHOD1(m,3) > 0 && METHOD1(m,3) < 180
 METHOD1(m,3) = METHOD1(m,3) + 167; % add 180 deg minus 13 deg magnetic
declination
 elseif METHOD1(m,3) > 180 && METHOD1(m,3) < 360
 METHOD1(m,3) = METHOD1(m,3)—193; % subtract 180 degrees and 13 deg
megnetic declination
 end


 METHOD1(m,4) = METHOD1(m,2)/METHOD1(m,1); %Calculate wind velocity
 METHOD1(m,5) = DATA(m,4)-280;
 end
end

windtime(1,1) = 0 + METHOD1(1,1);
for m = 2:length(METHOD1)
 windtime(m,1) = windtime(m-1,1) + METHOD1(m,1);
end

MeanWindMag = mean(METHOD1(:,4));
MeanWindDir = mean(METHOD1(:,3));

figure()
subplot(121)
plot(METHOD1(:,4),METHOD1(:,5),'*-');
title('360^o Method—Wind Magnitude Profile');
xlabel('Wind Magnitude (m/s)');
ylabel('Altitude AGL (m)');
grid on
hold on
plot([MeanWindMag MeanWindMag],get(gca,'ylim'),'r--');
Y=axis;
text(MeanWindMag,0.97*Y(4)+0.03*Y(3),['\leftarrowMean |W| = '
num2str(MeanWindMag,2) ' m/s'])
xlim([0 10]);
hold off
subplot(122)
plot(METHOD1(:,3),METHOD1(:,5),'*-');
title('360^o Method—Wind Direction Profile');
xlabel('Wind Direction (from) (^o)');
```

```
ylabel('Altitude AGL (m)');
grid on
hold on
plot([MeanWindDir MeanWindDir],get(gca,'ylim'),'r--');
text(MeanWindDir,0.97*Y(4)+0.03*Y(3),['Mean Wind Direction = '
num2str(MeanWindDir,3) '\circ'])
xlim([0 360]);
hold off



%% Plot showing the two heading sources
% Verify that the IMU and GPS headings are somewhat correlated. In most
% cases, the GPS lags the IMU while the IMU has more noise.
figure()
plot(UsefulData(:,2),UsefulData(:,10)); % IMU Hdg plot
hold on
plot(UsefulData(:,2),UsefulData(:,29)); % GPS Hdg Plot
legend('IMU Heading','GPS heading');
title('Comparison of Heading Sources');
xlabel('Time (s)');
ylabel('Heading (^o)');
hold off
%% METHOD 2: Using two Heading sources

%This method follows Oleg Yakimenko's derivation on p.403 of his text.
%Instead of using the circle rotations of METHOD 1, this method uses
GPS
%ground speed and GPS heading, along with IMU heading. The original
%derivation requires an airspeed measurement from a pitot tube.
Snowflake
%is not equipped with a pitot tube (one can be used for future works),
so
%the pitot velocity is fixed at 5 m/s for analysis purposes.

for m = 1:length(UsefulData)
 METHOD2(:,1) = UsefulData(:,2); % Time
 METHOD2(:,2) = UsefulData(:,3); % MSL Altitude (m)
 METHOD2(:,3) = UsefulData(:,27); % GPS Ground Speed
 METHOD2(:,4) = UsefulData(:,29); % GPS Heading
 METHOD2(:,5) = 3; % Pitot Airspeed (if available)
 METHOD2(:,6) = UsefulData(:,10); % IMU Heading data
 METHOD2(m,7) = METHOD2(m,3)*cos(degtorad(METHOD2(m,4)))...
-METHOD2(m,5)*cos(degtorad(METHOD2(m,6))); % Wind x-dir(North)
 METHOD2(m,8) = METHOD2(m,3)*sin(degtorad(METHOD2(m,4)))...
-METHOD2(m,5)*sin(degtorad(METHOD2(m,6))); % Wind y-dir(East)
 METHOD2(m,9) = sqrt(METHOD2(m,7)^2 + METHOD2(m,8)^2); % Wind Magnitude
 METHOD2(m,10) =
wrapTo360(radtodeg(atan2(METHOD2(m,8),METHOD2(m,7)))+167); %Wind
Direction "from"-magnetic variance
 METHOD2(:,11) = UsefulData(:,5); %Latitude
 METHOD2(:,12) = UsefulData(:,4); %Longitude
end
```

```
%% Remove all repetitive GPS data:
[NRD, ia, ic] = unique(METHOD2(:,4));
ia_sorted = sort(ia,'ascend');
counts = 1;
for mmm = 1:size(ia_sorted)
 NoRepeatData(mmm,:) = METHOD2(ia_sorted(mmm),:);
end
%% Calculate Mean Wind Magnitude and Direction

METHOD2MeanWindMag = mean(METHOD2(:,9));
METHOD2MeanWindDir = mean(METHOD2(:,10));
NoRepeatDataMeanWindMag = mean(NoRepeatData(:,9));
NoRepeatDataMeanWindDir = mean(NoRepeatData(:,10));

figure()
subplot(121)
plot(METHOD2(:,9),METHOD2(:,2)-280,'.');
title('Multi-Sensor—Wind Magnitude Profile');
xlabel('Wind Magnitude (m/s)');
ylabel('Altitude AGL (m)');
grid on
hold on
plot([METHOD2MeanWindMag METHOD2MeanWindMag],get(gca,'ylim'),'r--');
Y=axis;
text(METHOD2MeanWindMag,0.97*Y(4)+0.03*Y(3),['\leftarrowMean |W| = '
num2str(METHOD2MeanWindMag,2) ' m/s'])
xlim([0 15]);
hold off
subplot(122)
plot(METHOD2(:,10),METHOD2(:,2)-280,'.');
title('Multi-Sensor—Wind Direction Profile');
xlabel('Wind Direction (from) (^o)');
ylabel('Altitude AGL (m)');
grid on
hold on
plot([METHOD2MeanWindDir METHOD2MeanWindDir],get(gca,'ylim'),'r--');
text(METHOD2MeanWindDir,0.97*Y(4)+0.03*Y(3),['Mean Wind Direction = '
num2str(METHOD2MeanWindDir,3) '\circ'])
xlim([0 360]);
hold off

figure()
subplot(121)
plot(NoRepeatData(:,9),NoRepeatData(:,2)-280,'.');
title({'Multi-Sensor—Wind Magnitude Profile';'(No Repeat GPS Data)'});
xlabel('Wind Magnitude (m/s)');
ylabel('Altitude AGL (m)');
grid on
hold on
plot([NoRepeatDataMeanWindMag
NoRepeatDataMeanWindMag],get(gca,'ylim'),'r--');
Y=axis;
text(NoRepeatDataMeanWindMag,0.97*Y(4)+0.03*Y(3),['\leftarrowMean |W| =
' num2str(NoRepeatDataMeanWindMag,2) ' m/s'])
```

```
xlim([0 15]);
hold off
subplot(122)
plot(NoRepeatData(:,10),NoRepeatData(:,2)-280,'.');
title({'Multi-Sensor—Wind Direction Profile'; '(No Repeat GPS Data)'});
xlabel('Wind Direction (from) (^o)');
ylabel('Altitude AGL (m)');
grid on
hold on
plot([NoRepeatDataMeanWindDir
NoRepeatDataMeanWindDir],get(gca,'ylim'),'r--');
text(NoRepeatDataMeanWindDir,0.97*Y(4)+0.03*Y(3),['Mean Wind Direction
= ' num2str(NoRepeatDataMeanWindDir,3) '\circ'])
xlim([0 360]);
hold off

%% Plot Histograms of heading concentrations
y = 0:10:360;
mu1 = mean(NoRepeatData(1:425,10));
mu2 = mean(NoRepeatData(426:end,10));

sigma1 = std(NoRepeatData(1:425,10));
sigma2 = std(NoRepeatData(426:end,10));

f1 = exp(-(y-mu1).^2./(2*sigma1^2))./(sigma1*sqrt(2*pi));
f2 = exp(-(y-mu2).^2./(2*sigma2^2))./(sigma2*sqrt(2*pi));


figure()
subplot(121)
histogram(NoRepeatData(1:425,10),20);
title('Prevailing Wind Direction at High Altitude (>425m)')
xlabel('Heading (^o)');
ylabel('Number of Occurences');
subplot(122)
histogram(NoRepeatData(426:end,10),20)
title('Prevailing Wind Direction at Low Altitude (<425m)')
xlabel('Heading (^o)');
ylabel('Number of Occurences');

%% Roll and Pitch Oscillations vs Altitude
figure()
subplot(121)
plot(UsefulData(:,8)+90,UsefulData(:,3)-280); % Roll Plot
hold on
title('Roll and Pitch Oscillations vs Altitude')
plot([mean(UsefulData(:,8))+90
mean(UsefulData(:,8))+90],get(gca,'ylim'),'r--');
xlabel('Roll (^o)');
ylabel('Altitude AGL (m)');
grid on
axis tight
hold off
```

```
subplot(122)
plot(UsefulData(:,9),UsefulData(:,3)-280); % Pitch Plot
hold on
plot([mean(UsefulData(:,9)) mean(UsefulData(:,9))],get(gca,'ylim'),'r--
');
xlabel('Pitch (^o)');
ylabel('Altitude AGL (m)');
grid on
axis tight
hold off

%% Using Wind Estimate

%Assumptions:
% WIND: magnitude and direction are equal to the mean magnitude and
% direction and assumed constant from SF release until touchdown. Also,
% vertical winds are ignored (not the best practice at McMillan
Airfield
% due to mechanical wind produced on nearby hills.
% DENSITY: Air Density is assumed constant throughout the drop. Since
we
% are dropping from a relativeley low altitude, this assumption holds
for
% most cases.
% AIRCRAFT ATTITUDE: Straight and level, unaccelerated flight.
% INITIAL VELOCITY: Based on a fully deployed canopy ~5 m/s
% INITIAL HEDADING: Based off flight data from aircraft...parallel to
% runway is the standard at Camp Roberts during these tests.


v0h = 3; % Initial SF velocity after canopy filled
HDG0 = 280; % Initial Heading parallel to runway
TOF = UsefulData(end,2); %Time of Flight in seconds

W_North1 = MeanWindMag*cos(degtorad(MeanWindDir+180)); %N/S component
of wind velocity
W_East1 = MeanWindMag*sin(degtorad(MeanWindDir+180)); %E/W component of
wind velocity
W_North2 = METHOD2MeanWindMag*cos(degtorad(METHOD2MeanWindDir+180));
%N/S component of wind velocity
W_East2 = METHOD2MeanWindMag*sin(degtorad(METHOD2MeanWindDir+180)); %E/
W component of wind velocity

Initial_North = v0h*cos(degtorad(HDG0)); %SF initial N/S after canopy
inflation
Initial_East = v0h*sin(degtorad(HDG0));%SF initial E/W after canopy
inflation

SF_predicted_North_1 = ((W_East1 + Initial_East)*TOF)/1000;
SF_predicted_East_1 = ((W_North1 + Initial_North)*TOF)/1000;
magnitude_1 = sqrt(SF_predicted_East_1^2+SF_predicted_North_1^2); % in
km
SF_predicted_North_2 = ((W_East2 + Initial_East)*TOF)/1000;
SF_predicted_East_2 = ((W_North2 + Initial_North)*TOF)/1000;
```

```
magnitude_2 = sqrt(SF_predicted_East_2^2+SF_predicted_North_2^2); % in
km

% Predicted flightpath heading and distance
Flightpath_magnitude_1 = km2deg(magnitude_1);
Flightpath_Heading_1 =
radtodeg(atan2(SF_predicted_North_1,SF_predicted_East_1));
Flightpath_magnitude_2 = km2deg(magnitude_2);
Flightpath_Heading_2 =
radtodeg(atan2(SF_predicted_North_2,SF_predicted_East_2));

% Calculate Lat/Long for the predicted flightpath
[latout1,lonout1] =
reckon(UsefulData(1,5),UsefulData(1,4),Flightpath_magnitude_1,Flightpat
h_Heading_1);
[latout2,lonout2] =
reckon(UsefulData(1,5),UsefulData(1,4),Flightpath_magnitude_2,Flightpat
h_Heading_2);

% Create the birdseye plot again
figure()
if (CampRoberts == 1)
 CRImage = imread('CPRobertsflip','jpeg');
 DZimage = CRImage(:,:,1:3);
 image([-120.788473 -120.758492],[35.714764 35.731265], DZimage)
 axis([-120.788473 -120.758492 35.714764 35.731265])
 set(gca,'YDir','normal')
 daspect([1 cosd(UsefulData(end,5)) 1])
 hold on
end
plot(UsefulData(:,4),UsefulData(:,5),'.')
plot(UsefulData(1,4),UsefulData(1,5),'rs')
plot(UsefulData(end,4),UsefulData(end,5),'gd')

% Model 1 Visual Estimate
Vd=(UsefulData(1,3)-UsefulData(end,3))/(UsefulData(end,2)-
UsefulData(1,2)); %rate of descent
Times1=METHOD1(:,5)/Vd; % times to descent from 28 altitudes
Nblowaway1=Times1.*cosd(METHOD1(:,3)).*METHOD1(:,4)/110953+DATA(2:end,2
);
Eblowaway1=Times1.*sind(METHOD1(:,3)).*METHOD1(:,4)/90483+DATA(2:end,3)
;

 plot(Eblowaway1(14:end,1),Nblowaway1(14:end,1),'d-.')
%Landing
plot(Eblowaway1(end,1),Nblowaway1(end,1),'Marker','v','MarkerFaceColor'
,'red','MarkerSize',9);
%Method 1 Landing Error
[N2, E2] = geodetic2ned(UsefulData(end,5), UsefulData(end,4), 0,
Nblowaway1(end,1), Eblowaway1(end,1), 0, SPHEROID);
land_error = norm([N2,E2]);

% Model 2 (No Repeat) Visual Estimate
Times2 = NoRepeatData(:,2)/Vd;
```

```
Nblowaway2=Times2.*cosd(NoRepeatData(:,10)).*NoRepeatData(:,9)/110953+N
oRepeatData(:,11);
Eblowaway2=Times2.*sind(NoRepeatData(:,10)).*NoRepeatData(:,9)/90483+No
RepeatData(:,12);
plot(Eblowaway2(700:end,1),Nblowaway2(700:end,1),'d-.')
plot(Eblowaway2(end,1),Nblowaway2(end,1),'Marker','v','MarkerFaceColor'
,'blue','MarkerSize',9);

%Method 2 Landing Error
[N3, E3] = geodetic2ned(UsefulData(end,5), UsefulData(end,4), 0,
Nblowaway2(end,1), Eblowaway2(end,1), 0, SPHEROID);
land_error2 = norm([N3,E3]);

h=legend('Trajectory','Canopy deployment','Touchdown','360 Turn Method
Flight Path','360 Turn Method Estimated Landing','Multi-Sensor Method
Flight Path','Multi-Sensor Method Estimated
Landing','location','best'); set(h,'fontsize',8);
xlabel('Latitude (^o)');
ylabel('Longitude (^o)');
axis tight
title('Wind Estimation Example');
```

LIST OF REFERENCES

[1] Institute of Electrical and Electronics Engineers. “Playing with parachutes.” (n.d.). [Online]. Available: http://tryengineering.org/lessons/playingwithparachutes.pdf. [Accessed May 15, 2016].

[2] History Channel. “This day in history.” (n.d.). [Online]. Available: http://www.history.com/this-day-in-history/the-first-parachutist. [Accessed Jul 16, 2016].

[3] P. A. Combes, “Calculation of the wind profile and its effects on a flight of an Aerial Delivery System,” unpublished.

[4] Should the atomic bombs have been used on Japan? (2015, Dec.). Penn State University. [Online]. Available: http://sites.psu.edu/worldwar2ending/operation-downfall.

[5] O. Yakimenko, Ed., Precision Aerial Delivery Systems: Modeling, Dynamics, and Control, Reston, VA: American Institute of Aeronautics and Astronautics, Inc., 2015.

[6] J. S. Lindegard, “Precision aerial delivery seminar: ram-air parachute design,” presented at the 13th AIAA Aerodynamic Decelerator Systems Technology Conference, Clearwater Beach, FL,1995.

[7] Department of Defense, Joint Precision Airdrop Systems (JPADS) Programs: Precision, Never Late Logistics for the Future Force, Natick: U.S. Army Natick Soldier RD&E Center, 2007.

[8] M. R. Driels, Weaponeering: Conventional Weapon Systems Effectiveness, 2nd ed., Reston: American Institute of Aeronautics and Astronautics, 2013.

[9] M. Sadraey, “Wing Design.” (n.d.). Daniel Webster College, [Online]. Available: http://faculty.dwc.edu/sadraey/Chapter%205.%20Wing%20Design.pdf. [Accessed May 12, 2015].

[10] 3DR. “3DR Pixhawk.” (2016). [Online]. Available: https://store.3dr.com/products/3dr-pixhawk.

[11] ArduPilot Dev Team. “Mission Planner Overview.” (2016). [Online]. Available: http://ardupilot.org/planner/docs/mission-planner-overview.html.

[12] M. Ryan. (2012). “Ryan Mecatronics.” [Online]. Available: http://www.ryanmechatronics.com/XMonkey.htm.

[13] Arcturus UAV. “T-20.” (2015).[Online]. Available: http://arcturus-uav.com/product/t-20.

[14] J. B. Kuipers, Quaternions and Rotation Sequences: A Primer with Applications to Orbits, Aerospace, and Virtual Reality, Princeton: Princeton University Press, 1999.

[15] Center for Interdisciplinary Remotely-Piloted Aircraft Studies. “McMillan facilities.” (n.d.). [Online]. Available: http://www.cirpas.org/facilities.html. [Accessed August 16, 2016].

[16] Federal Aviation Administration. “Airspace, Special Use Airspace, and TFRs.” (2016). [Online]. Available: https://www.faasafety.gov/gslac/ALC/course_content.aspx?cID=42&sID=242&preview=true.

[17] McMillan Airport (CA62). (2016, Jul. 21). AirNav. [Online]. Available: https://www.airnav.com/airport/CA62.

[18] Google maps—McMillan, Paso Robles, CA. (2012). AirNav.com. [Online]. Available: https://www.google.com/maps/place/Mc+Millan/@35.7189117,-120.7811392,2972m/data=!3m1!1e3!4m5!3m4!1s0x80ecd3b747092127:0xdd68169c4f5a48fe!8m2!3d35.7180039!4d-120.7676804?hl=en.

[19] J. Osborne and R. Rysdyk, “Waypoint guidance for small UAVs in wind,” Autonomous Flight Systems Laboratory, Arlington, VA, AIAA 2005-6951, Sep. 2005.

[20] B. H. J. Palanthandalam-Madapusi, A. Girard, and D. Bernstein, “Wind-field reconstruction using flight data,” presented at American Aeronautical Control Conference, Seattle, WA, 2008.

[21] MathWorks. “The language of technical computing.” (2016). [Online]. Available: http://www.mathworks.com/products/matlab/.

[22] C.W. Hewgley IV. “Pose and wind estimation for autonomous parafoils,” Ph.D. dissertation, Dept. of Electrical Engineering, Naval Postgraduate School, Monterey, CA, 2014.

INITIAL DISTRIBUTION LIST

1. Defense Technical Information Center
   Ft. Belvoir, Virginia

2. Dudley Knox Library
   Naval Postgraduate School
   Monterey, California